PSYCHOLOGICALLY CONSISTENT COORDINATED CONTROL OF MULTI-AGENT TEAMS

A Dissertation
Presented to
The Academic Faculty

By

Tina Setter

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2017

PSYCHOLOGICALLY CONSISTENT COORDINATED CONTROL OF MULTI-AGENT TEAMS

Approved by:

Dr. Magnus Egerstedt, Advisor
Professor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Fumin Zhang
Associate Professor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Yorai Wardi
Professor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Sonia Chernova
Assistant Professor
School of Interactive Computing
Georgia Institute of Technology

Dr. Ayanna Howard
Professor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: March 28, 2017

To my mom. Thank you for always being there.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Magnus Egerstedt, to whom I would not have survived this process without. He has not only helped me grow immensely as a researcher, but also as a teacher and as a person. I am very grateful to have had his invaluable guidance throughout my journey here at Georgia Tech. Moreover, I would like to thank Dr. Yorai Wardi and Dr. Ayanna Howard for graciously agreeing to be on my reading committee and Dr. Fumin Zhang and Dr. Sonia Chernova for also giving their time to be on my dissertation committee.

Much of my work would not have been possible without the collaborations that I have had with other people in the field. I want to thank Dr. Andrea Gasparri for his input and guidance over the past couple of years on our human-robot trust research. I would also like to acknowledge Dr. Hiroaki Kawashima and Alex Fouraker for their efforts on the haptic manipulability work. Additionally, I am thankful for the many helpful discussions I have had with the other members of the GRITS Lab, both in the lab and out.

It would be difficult to go through this process without a support system, and I am extremely grateful to have made many close friends while I have been at Georgia Tech. I want to thank my "school" family – Stephanie, Michael, Andy, Tom, and Peter – for always helping me out, whether they were enduring my practice presentations or providing a break from studies through board game nights. I would also like to thank my good friend Kristen, a fellow Ohio State alumna, for always being there for me and making Georgia feel a little bit more like home.

Last but certainly not least, I would like to thank my mom, who has never stopped believing in me throughout not only my Ph.D., but my entire life. I would not be where I am today if it was not for her constant love, support, and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this research is to describe both human-robot interactions and inter-robot interactions and analyze the behavior of the resulting multi-agent systems, while drawing comparisons to psychological studies regarding human team behavior. In particular, we look at the effects of trust, energy, and manipulability on these interactions. We first address the problem of modeling trust evolution and describing how it affects the states of agents in a system – whether they be human or robot. We introduce two different types of trust models – self-centered and team-oriented – and show, through simulations and theoretical analyses, under what initial trust conditions these systems achieve their objectives. We show our models to be psychologically consistent in that they exhibit group polarization, belief polarization, and a positive trust-performance correlation.

In the second part of this work, we look at the effect of energy on inter-robot interactions by solving an energy-constrained coordination problem in which robots must determine where and when to meet given differing initial battery levels to do so in the least amount of time. This is formulated as a constrained optimization problem where the constraints arise from solving for a single agent's optimal control input. Lastly, we address the effect of manipulability on human-robot interactions through a haptic human-swarm interaction user study. Manipulability, a notion describing how effective a leader robot is at controlling the follower robots, is provided as force feedback on a haptic joystick that a human operator uses to control a swarm of robots. Ten subjects complete the experiment in which they move the group of robots through a series of waypoints and different mappings between manipulability and the haptic feedback force are investigated.

CHAPTER 1

INTRODUCTION

The literature regarding distributed control algorithms for multi-robot systems is abundant, where local interactions between robots are designed to yield global behaviors. However, it is unclear what happens when humans are introduced into these multi-robot teams and what factors may affect the interactions between humans and robots. We investigate this question, specifically focusing on trust and how it affects these human-robot interactions. And, expecting that the way in which people behave in teams with other people will have some bearing on how they behave in teams with robots, we draw upon a rich body of psychology literature regarding human-human interactions in teams in order to validate our work. The objective of this research is to describe both human-robot interactions and inter-robot interactions and analyze the behavior of the resulting multi-agent systems, while ensuring that the results are consistent with psychological studies regarding human teams.

The area of multi-agent robotics has grown in popularity in the last decade, as the idea of deploying many robots has far-reaching benefits and applications. One main benefit of multi-agent systems is that they are robust to failures, that is, if one robot fails, there are plenty of robots remaining to complete the task. Additionally, these robots can often be made smaller, less complex, and thus ultimately less expensive than one single robot designed to do the same task. This also makes them able to be replaced more easily when there is a failure. There are numerous applications for multi-agent systems that have been alluded to in the literature that will be discussed in Section 2.1.

As robots become increasingly present in our daily lives, it is important that we understand how people and robots will coexist and interact. The area of human-robot interaction (HRI) focuses on this expected teaming, with research questions spanning a broad spectrum of topics. Furthermore, human-swarm interaction (HSI), a field devoted to studying the interactions between one human and a group of robots, is emerging with the rising interest in

incorporating human input into teams of robots, for applications such as search and rescue and area surveillance. A representative sample of the current research in these areas will be discussed in Section 2.2.

Of the many factors that come into play when humans interact with robots, trust is arguably one of the most important. It was shown in [1] that a lack of trust may make people less willing to accept information provided by a robot and thus they will not maximally benefit from the advantages that are typically present in a robotic system. Successful HRI relies on creating appropriate levels of trust and much attention has been made to determining the factors that affect trust. Although the research in this area has mostly focused on one human-one robot interaction scenarios, our goal is to determine how trust affects human-swarm interaction scenarios. To this point, however, the work done on analyzing trust in human-swarm interactions has been limited and in the work that does exist, the human is often viewed as an operator. Instead of viewing the human as an operator, we are interested in human-robot teaming where, not only can the humans assist the robots, but the robots can also provide useful information to the humans. A survey of the literature regarding trust is in Section 2.3.

The main contribution of this thesis is in Chapter 4, where we aim to understand the effects of trust on the performance of human-robot teams by combining standard multi-agent control laws with carefully designed trust dynamics. We introduce two different trust models – *self-centered* and *team-oriented* – and describe mathematically how these trust metrics affect the agents' states. For both of these models, the behavior of the resulting coupled system is analyzed and conditions under which it yields the desired performance are presented. In addition to allowing all agents to have the same dynamics, we also present models with heterogeneous dynamics, where human agents possess trust states but robot agents do not, and analyze the asymptotic behavior of these systems as well. We also compare the convergence rate of the system with trust to that of the standard consensus protocol

as well as the point at which they converge. Because human teams have been studied extensively in psychology, we make comparisons between our results and phenomena that have been observed for groups consisting solely of humans.

In the trust-based algorithms, if the goal of the agents is to meet at some common spatial location, initial trust is shown to affect the location where the agents end up meeting. In Chapter 5, we explore a similar situation where, instead of trust, energy is the metric that potentially changes the location where agents meet during rendezvous. Many robotic applications, e.g. environmental monitoring, involve sending robots out for long periods of time, and it is important to take energy into account when designing coordination algorithms for such systems. In this work, we determine where and when a group of mobile robots should meet, given that they need to do so in the least amount of time and taking into consideration that they all have different initial battery levels. This is formulated as a constrained optimization problem, where the constraints ensure that each robot can reach the rendezvous point in the specified time given its available battery life. The algorithm is implemented on a team of differential drive robots to show its practicality.

In previous chapters, the focus is on additional states that impact the behavior of coordinated teams, namely trust and energy. In Chapter 6, we change perspective slightly and investigate a more practical matter regarding how humans could engage with multi-robot teams. In particular, we employ a haptic human-swarm user study to determine the best way to relay information about the state of a robotic swarm to a human operator. Manipulability, a notion describing how effectively a leader robot is controlling a group of follower robots, is used to inform a human operator about how well he or she is controlling a group of robots. Ten subjects perform an experiment in which they control the leader robot's velocity with a joystick while the manipulability information is fed back through haptic forces on the joystick. We investigate different mappings between manipulability and the haptic feedback force to determine which one is most effective in terms of team performance and operator workload.

CHAPTER 2

LITERATURE REVIEW

The work in this thesis primarily falls within the intersection of three fields: multi-agent systems, human-robot interaction, and trust-based algorithms. In this chapter we give an overview of the existing literature in these areas and explain how our work differs.

2.1 Multi-Agent Robotics

The field of multi-agent robotics has matured significantly over the last decade, as the idea of deploying many small robots as opposed to one large robot has far-reaching benefits and applications. One main benefit is that of robustness to failures; that is, if one robot fails, there are many remaining that are capable of completing the task. Also, robots used in multi-agent scenarios are typically smaller and less expensive than one large complex robot designed for the same task. Many applications for these systems have been explored in the literature, including space exploration [2], military missions [3], and search and rescue [4], to name a few.

As the focus is on having large numbers of these robots, it is important that the control algorithms be decentralized. In other words, the agents should be able to use only local interaction rules to produce global behaviors. By making the interactions local, as opposed to requiring centralized computation, the algorithms can be scaled up arbitrarily largely. Some thoroughly investigated distributed multi-agent algorithms include consensus [5, 6], area coverage [7, 8], formation maintenance [9, 10], flocking and swarming [11], and containment control [12].

In this thesis, we will be mostly referring back to the problem of consensus, in which a group of agents are tasked with agreeing on some value, e.g. an opinion or a physical property, such as a spacial location. More details regarding this canonical multi-agent problem will be presented in Chapter 3.

2.2 Human-Robot Interaction

On their own, multi-robot systems are useful, but designing coordination algorithms for these robots becomes more challenging when they need to operate in unknown or dynamically changing environments. When high-level decisions need to be made on the fly, the cognition and reasoning ability of a human can be extremely beneficial. Moreover, as robots start appearing on our roads and in our households, it will become inevitable that robots and humans interact. Human-robot interaction and human-swarm interaction are two fields that help us understand and design for these scenarios in which robots and humans need to cooperate.

Human-swarm interaction (HSI) is becoming increasingly important and is accordingly receiving greater attention in the past decade (see [13] for a survey of the literature), and while studies have shown experimentally what these interactions might look like for specific applications or using specific interaction modalities [14–16], we are far from an all-encompassing theory of HSI. And, in order to make progress towards this theory, work must be done on modeling factors that affect the interactions that occur when humans are injected into multi-robot teams. The factor that we are primarily interested in is trust.

As noted previously, [1] showed that human trust in a robotic partner plays an important role in the outcome of the interaction because it affects how willing the human is to accept information provided by the robot. It has also been stated that appropriate levels of trust are crucial for successful human-robot interactions, which has been shown to be challenging to ensure [17]. Because of this, much attention has been made to determining the factors that affect trust in HRI [18, 19].

As trust has been shown to play a large role in human-robot interactions, it is clear that this importance extends to human-swarm interactions [20, 21]. To this point, however, the work done on analyzing trust in human-swarm interactions has been limited and in the work that does exist, the human is often viewed as an operator. For example, in [22], trust is used to schedule the attention of a human operator between multiple robots and in [23],

trust is used to blend commands from a human operator with those from an autonomous controller commands to achieve higher performance when teleoperating mobile robots. In [24], a system dynamics model was developed to analyze the impact of operator trust on performance in multiple robot control.

Up to this point, much of the work done on trust in human-swarm interaction has been anecdotal and based on user studies. The work that we will present in Chapter 4 aims to understand the effects of trust on human-robot teams through rigorous theoretical analysis. And, in order to do this analysis, we first need to develop models of trust evolution and connect this trust metric to the behavior of the teammates. In the next section we will review some of the related efforts that have been made in this area.

2.3 Trust-Based Algorithms

In most of the multi-agent literature, agents are assumed to be working cooperatively towards the same goal. For example, under the consensus protocol, each agent updates its state to be closer to the other agents' states. However, in some cases, there may be agents that are malicious or not working towards the same goal as the rest of the team. In these cases, an agent may want to weigh the information from its neighbors according to which agents it "trusts" the most. This idea is inspired by human group collaboration, where people tend to place a higher importance on the opinion of those that they trust. Opinion dynamics, a field concerned with understanding how peoples' opinions are affected by those with whom they interact, has been studied extensively, e.g., [25–27], and the idea of incorporating trust grew from the bounded confidence models of Deffuant [28] and Hegselmann and Krause [29].

Trust-based multi-agent algorithms can be broken down into two main components: the coupling between trust and agent dynamics and the trust model. First we will discuss the coupling between the trust and the agents' state dynamics. If we let τ_{ij} represent how much agent i trusts agent j, then the coupling typically comes in the form of a weighted

consensus equation, that is

$$\dot{x}_i = \sum_{j \in N_i} \tau_{ij}(x_j - x_i), \tag{1}$$

where N_i is the set of agents with whom agent i shares information. Here, agent i's state is influenced more heavily by the agents that it trusts more. This coupling has been explored in the multi-agent controls community, in the context of opinion forming.

In [30], the authors represent antagonistic interactions with negative weights, that is $\tau_{ij} < 0$, and they show that under certain conditions, agents can still achieve so-called modulus consensus, where all agents' opinions converge to the same magnitude, but may have the opposite sign. In [31], the same state dynamics are used and cases in which the system can achieve a unanimous opinion are found, which is described as a consensus on the signs of the opinions. However, the trust in these works is represented by static weights, whereas we are interested in trust that evolves as a function of how the agents are behaving.

Similar approaches were taken in [32–34] by incorporating time-varying weights (i.e., τ_{ij} is a function of time). These weights could be used to represent trust between agents, but they are not dynamic in the sense that they do not depend on the agents' states and thus the trust does not reflect the dynamic behavior of the agents. In fact, [30–34] do not describe a model for trust evolution whatsoever.

Several trust models have been developed, both in the controls community and in the computing community. In the areas of large-scale open distributed computing and network security, trust is often derived from a combination of an agents' opinion and the opinions of its neighbors, and history of the trust is often taken into account as well. The work in [35–37] uses these kinds of trust models in order to mitigate effects of malicious nodes in computing applications. They do not, however, describe how the trust is coupled to the agents' state dynamics.

In the physics and controls fields, the Hegselmann-Krause (HK) model has been explored extensively, e.g., in [38–42], in which two agents interact only if their belief states are sufficiently close. In other words, two agents only take into account each other's opinions

if their states differ by less than some value ϵ, and do not interact otherwise. That is,

$$\tau_{ij} = \begin{cases} 1, & \|x_j - x_i\| < \epsilon \\ 0, & otherwise \end{cases}$$

where the state dynamics are given in (1). In [43], several trust models are explored, where they build on the HK model by incorporating probability, history, and reputation, as is done in much of the computing literature. The work in [44] also uses trust models that take history and reputation into account but they are more general in the sense that they utilize a general mapping of a feature space to favorable/unfavorable regions in order to determine trustworthiness and give results that do not depend on the choice of mapping.

Other state-dependent trust models have been developed for specific applications, such as coverage [45] and multi-robot patrolling [46]. In these works, the trust changes in response to performance variations among the robots, where the performance metric is the quality of sensor measurements in the coverage case and the maximum refresh time in the patrolling scenario. In both of these examples, the trust values affect the way that the agents execute their mission, and thus has some coupling to the agent dynamics, but it is not in as simple of a manner as in (1). In these situations, trust is determined by a more complex question of whether an agent is behaving as it is expected to, as opposed to in the HK trust model, which only considers how close agents' opinions are to each other.

To summarize, we need both a trust model that describes how the inter-agent trust evolves (could be performance- or state- dependent) and a coupling between the trust and the agents' dynamics, which describes how the trust affects the multi-agent coordination process. Not much has been done in terms of state-dependent, continuous trust values, and thus we focus our work in this realm, while also ensuring that we have a tight coupling between the trust and the agent dynamics. We use the agent state dynamics in (1), but derive the trust dynamics differently than in the works discussed here. We provide a general model for trust in multi-agent systems, which is derived using a cost function that represents some overall objective of the team. This work will be discussed in Chapter 4.

CHAPTER 3

PRELIMINARIES

Because the work in Chapters 4 and 5 present variations of the canonical multi-agent consensus problem, we will present a detailed description of this problem and the proposed solution in the literature. Before doing so, some preliminaries regarding graph theoretic notions for multi-agent systems are given. These notions will be used repeatedly throughout the thesis as our focus is on multi-agent systems that have a specific interaction structure that can be represented using a graph theoretic framework.

3.1 Multi-Agent Graph Theoretic Framework

When designing control algorithms for multi-agent systems, it is desirable that the agents need only to interact locally with a subset of the other agents, as opposed to having to interact with everyone. The agents that a particular agent can share information with are deemed their neighbors and any two neighboring agents are said to be *adjacent*. As is done repeatedly in the literature, e.g., [8, 10, 47], the interaction topology can be represented using a graph $G = (V, E)$, where the *vertex set* $V = \{v_1, v_2, \ldots, v_N\}$ is the set of nodes, where N is the total number of agents, and the *edge set* $E \subseteq V \times V$ is a set of pairs indicating which agents can share information. That is, $(v_i, v_j) \in E$ if and only if nodes v_i and v_j are adjacent. Throughout this thesis, pairs in the edge set may also be referred to by (i, j) to simplify notation.

The *neighborhood* $N_i \subseteq V$ of vertex v_i is defined as the set $\{v_j \in V \,|\, (v_i, v_j) \in E\}$. In other words, this is the set of all vertices that are adjacent to v_i. In an *undirected* graph, $v_i \in N_j$ implies that $v_j \in N_i$, or that the set E consists of unordered pairs. A *path* of length m in G is defined to be a sequence of distinct vertices

$$v_{i_0}, v_{i_1}, \ldots, v_{i_m}$$

such that $(v_{i_k}, v_{i_{k+1}}) \in E$ for $k = 0, 1, \ldots, m-1$. In the sequence, v_{i_0} and v_{i_m} are the end

vertices of the path and the rest are the inner vertices.

The graph G is called *connected* if there exists a path between every pair of vertices in V. Otherwise, the graph is called *disconnected*. One standard graph that comes up often in the literature is the *complete graph*, denoted by K_N where N is the number of vertices. This graph is characterized by every vertex in V being adjacent to every other vertex in V.

One way that a graph G can be described is using a matrix representation, known as is its *graph Laplacian*, $L(G) = [\ell_{ij}]$. It is defined as

$$\ell_{ij} = \begin{cases} -1, & v_j \in N_i \\ |N_i|, & j = i \\ 0, & otherwise. \end{cases}$$

This will become important later when we discuss the convergence properties of multi-agent algorithms. With these graph theoretical notions in mind, we now present a canonical multi-agent coordination problem, consensus.

3.2 Multi-Agent Consensus

The research presented in this thesis will often focus on variations of the consensus problem, in which a group of agents are tasked with agreeing on some value, e.g., an opinion or a physical property, such as the agent's position. The standard consensus algorithm, e.g. in [48], consists of each agent updating its own state by taking a weighted average of its own and its neighbors' states. Agent i's state is given by $x_i \in \mathbb{R}$, N_i is the set of indices corresponding to the agents with whom agent i can share information and N is the total number of agents. We focus on one-dimensional consensus here for notational simplicity, but it should be noted that this algorithm can be scaled to higher dimensions by performing consensus over each dimension. Agent i's state is updated according to the consensus protocol, given by

Consensus Protocol

$$\dot{x}_i = \sum_{j \in N_i} (x_j - x_i). \tag{2}$$

If the states are stacked into a vector $x = [x_1, x_2, ..., x_N]^T$, then the ensemble-level dynamics can be written as

$$\dot{x}(t) = -Lx(t)$$

where L is the aforementioned graph Laplacian. The ensemble form of the dynamics makes this system easier to analyze and convergence results are shown in [48]. This protocol indeed results in the agents reaching consensus, provided that the underlying information-exchange graph G is connected. That is, the states converge to a configuration such that $x_i = x_j$ for all i, j.

Moreover, we know exactly *where* the states converge. As shown in [47], if G is connected, then

$$x(t) \to \frac{\mathbf{1}^T x(0)}{N} \mathbf{1} \text{ as } t \to \infty,$$

where $\mathbf{1} \in \mathbb{R}^N$ is a column vector of all 1's. In other words, each component of x converges to the mean of the initial states. Also it should be noted that the rate of convergence is dictated by the smallest positive eigenvalue of the graph Laplacian L. These results will become especially important in Chapter 4 when we compare the asymptotic behavior of our trust-based interaction models to that of the standard consensus protocol.

CHAPTER 4

TRUST-BASED INTERACTIONS IN HUMAN-ROBOT TEAMS

Although trust is most commonly thought of in the context of human-human interactions, it is important to understand how it affects human-robot and robot-robot interactions. Not only could this help in mitigating the effects of malicious agents, it also gives us a framework by which we can analyze teams consisting of humans and robots that complete tasks collaboratively. In this chapter, we introduce two types of trust models: self-centered and team-oriented, and show how this trust is connected to the behavior of the members of the team, whether robot or human. We then analyze the behavior of the systems formed by these models and show under what conditions the team will achieve its objective.

We present the self-centered trust model in Section 4.1 and discuss invariance and convergence results of the corresponding coupled trust-state system. In Section 4.2, we discuss similar invariance and convergence analyses, but for the team-oriented trust model. For both models, we also introduce a system with heterogeneous dynamics which is intended to model human-robot teams by allowing humans and robots to have different behaviors. In Section 4.3 we explore the idea of trust-action couplings, in which collective decisions result in agents taking actions, which in-turn changes the dynamics of the system depending on trust levels.

The majority of the work in this chapter has been, or will be, published in [49] and [50] and is under review in [51].

4.1 Self-Centered Trust

Before describing the model, we first give a two-agent example to illustrate how we define trust and connect it to state evolution.

4.1.1 A Two-Agent Mood Picture

The reason why it is both problematic and worth-while to explicitly couple the agents' opinions about neighboring agents to their actions is that these types of coupled interaction

effects (through both beliefs and physical states) will inevitably play some parts in future human-swarm interaction scenarios. But, these couplings are indeed quite delicate in that a lot of surprising effects can emerge. As a first illustration of this, as a cautionary tale, consider the simplest possible case of two agents with scalar states $x_i,\ i = 1, 2$, who are to meet at a joint location, i.e., by solving the, by now, classic rendezvous problem [52].

Letting the quantity $\frac{1}{2}(x_1 - x_2)^2$ be the measure that the agents wish to collectively minimize, we note that the change in performance is given by

$$\frac{d}{dt}\left(\frac{1}{2}(x_1 - x_2)^2\right) = (x_1 - x_2)\dot{x}_1 + (x_2 - x_1)\dot{x}_2.$$

As the term $(x_1 - x_2)\dot{x}_1$ encodes how much the movement of agent 1 contributes to the change in the performance (and vice versa for agent 2), the trust that agent 2 "feels" towards agent 1 should reflect this fact. In other words, if agent 2 is contributing a lot to the agents getting closer, then agent 1 should trust agent 2 more.

A possible encoding of the previous observation could be to let the scalar trusts τ_1 and τ_2 evolve as

$$\dot{\tau}_1 = -(x_2 - x_1)\dot{x}_2, \quad \dot{\tau}_2 = -(x_1 - x_2)\dot{x}_1,$$

where the negative sign is used to describe the fact that a reduction in inter-agent distance should correspond to an increase in trust. Moreover, the trust itself needs to be coupled to the motion of the two agents. And, following the observation of group and belief polarization, we let a positive trust τ_1 mean that the Agent 1 is indeed moving towards Agent 2, while a negative trust would mean the opposite. In the context of the consensus equation (e.g., [47]), this could be directly encoded using the trusts as weights, i.e.,

$$\dot{x}_1 = \tau_1(x_2 - x_1), \quad \dot{x}_2 = \tau_2(x_1 - x_2).$$

Substituting these expressions for $\dot{x}_i,\ i = 1, 2$, into the trust update laws thus yields the composite system

$$\begin{aligned} \dot{x}_1 &= \tau_1(x_2 - x_1) & \dot{x}_2 &= \tau_2(x_1 - x_2) \\ \dot{\tau}_1 &= \tau_2(x_1 - x_2)^2 & \dot{\tau}_2 &= \tau_1(x_1 - x_2)^2, \end{aligned} \tag{3}$$

which, due to the presence of square terms, is not globally Lipschitz, i.e., there might even be issues with the existence of solutions [53].

In fact, if we let $\xi = x_1 - x_2$ denote the disagreement between the two agents, and $\hat{\tau} = \tau_1 + \tau_2$ the total trust in the system, we note that $\dot{\xi} = -\hat{\tau}\xi$ and $\dot{\hat{\tau}} = \hat{\tau}\xi^2$. From these equations we note that

$$\dot{\hat{\tau}} = -\xi\dot{\xi} \;\Rightarrow\; \hat{\tau} = -\frac{1}{2}\xi^2 + c,$$

for some constant c, and therefore the expression $\hat{\tau} + 1/2\xi^2$ is invariant with respect to time. This fact will carry over to more complex models as will be seen for the general case and we state this observation as a lemma.

Lemma 1. *Under the two-agent dynamics given in* (3)*, the following holds*

$$\frac{d}{dt}\left(\hat{\tau} + \frac{1}{2}\xi^2\right) = 0.$$

First we would like to determine when this system will result in the agents' states converging to the same point, and when it will cause the states to diverge. And, since $\xi(0) = 0$ implies that an agreement is trivially reached initially – and maintained throughout – it is assumed, for the remainder of this section, that $\xi(0) \neq 0$.

The first thing to note about this system is that if $\hat{\tau} > 0$, then $\dot{\hat{\tau}} > 0$ as long as $\xi \neq 0$. This means that if $\hat{\tau}(0) > 0$, the total trust in the system will increase monotonically. But the invariance $\xi^2 = 2(c - \hat{\tau})$ implies that this increase in $\hat{\tau}$ will have to correspond to a decrease in ξ. In other words, if $\hat{\tau}(0) > 0$ then $\hat{\tau}$ will increase monotonically until, in the limit, $\xi = 0$, i.e., the two agents will indeed agree asymptotically.

Similarly, if the initial, total trust is 0, then both $\dot{\hat{\tau}}$ and $\dot{\xi}$ are equal to zero and the initial disagreement does not change, although both x_1 and x_2 do change at the same rate. We would additionally like to understand what happens if instead, the initial, total trust is less than 0. We explore this by setting

$$\eta = \frac{1}{\xi^2}$$

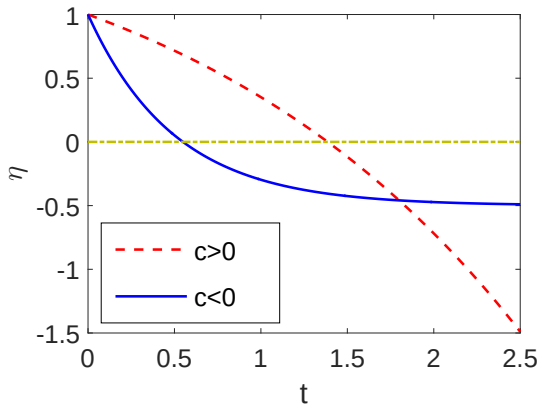


Figure 1. The two cases of $c > 0$ and $c < 0$ are shown when the initial, total trust is negative. The case $c > 0$ causes η to decay asymptotically to $-\infty$ and $c < 0$ causes η to converge to a negative value. In both cases, η crosses zero, causing ξ to exhibit finite escape time.

which results in the solution (assuming $c \neq 0$)

$$\eta(t) = e^{2ct}\left(\eta(0) - \frac{1}{2c}\right) + \frac{1}{2c}.$$

We now note that $\eta(0) > 0$ (as long as $\xi(0) \neq 0$) and two different cases need be addressed, namely when $c > 0$ and when $c < 0$. If $c < 0$, $\eta(t)$ will decay exponentially from $\eta(0) > 0$ to $1/(2c) < 0$, and thus cross $\eta = 0$ at some finite time. But, since $\eta = \xi^{-2}$, this implies that $\xi(t)$ exhibits finite escape time, i.e., it goes to $\pm\infty$ in finite time.

If $c > 0$ we note that $\hat{\tau}(0) < 0$ implies that

$$\hat{\tau}(0) = -\frac{1}{2}\xi^2(0) + c < 0 \;\Rightarrow\; \eta(0) < \frac{1}{2c} \;\Rightarrow\; \eta(0) - \frac{1}{2c} < 0.$$

In this case, η starts at $\eta(0) > 0$ and then decays exponentially to $-\infty$. As such, there exists a finite time at which $\eta = 0$, i.e., also in this case does $\xi(t)$ exhibit finite escape time. As a final note, if $c = 0$ then $\eta(t) = -t + \eta(0)$, i.e., at time $t = \eta(0)$, the error dynamics escapes to $\pm\infty$.

These two cases are shown in Figure 1, and We have thus established the following two-agent theorem:

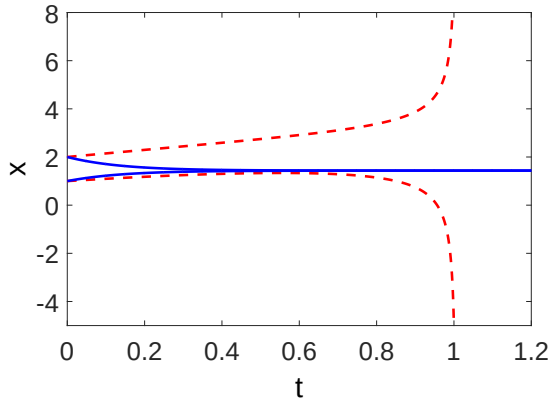


Figure 2. Two different scenarios are shown. The first (middle trajectories) corresponds to the total, initial trust being positive, causing the two agents to reach an agreement asymptotically. The second case corresponds to a negative initial trust, resulting in diverging states in finite time.

Theorem 1. *Consider the two-agent system*

$$\dot{x}_1 = \tau_1(x_2 - x_1) \qquad \dot{x}_2 = \tau_2(x_1 - x_2)$$
$$\dot{\tau}_1 = \tau_2(x_1 - x_2)^2 \qquad \dot{\tau}_2 = \tau_1(x_1 - x_2)^2.$$

If $\tau_1(0) + \tau_2(0) > 0$ *then* $\lim_{t\to\infty} |x_1(t) - x_2(t)| = 0$. *If* $\tau_1(0) + \tau_2(0) = 0$ *then* $x_1(t) - x_2(t)$ *is constant. Finally, if* $\tau_1(0) + \tau_2(0) < 0$ *then* $|x_1(t) - x_2(t)|$ *diverges to infinity in finite time, whenever* $x_1(0) \neq x_2(0)$.

These phenomena are shown in Figure 2, where two cases are shown; one where $\hat{\tau}(0) > 0$, and the agents agree asymptotically, and one where $\hat{\tau}(0) < 0$, and they diverge in finite time.

Theorem 1 tells us that through the addition of an innocent-looking trust dynamics that is coupled to the agents' state update laws, not only can the system diverge, it may even diverge in finite time. This means that if two agents do not trust each other sufficiently much initially, the process deteriorates completely. If, for example, one were to add a cut-off, as is done in Krause's model, i.e., the agents only take each other into account if their distrust is not too great, polarized states are achieved that are moreover more extreme than the agents' initial states, as shown in Figure 3. That is, belief polarization results.

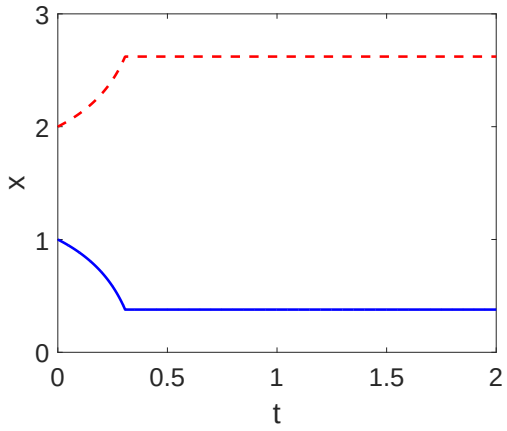


Figure 3. Belief polarization is achieved with the agents assuming more extreme positions than their initial positions by only paying attention to each other if the trust values are above a certain threshold.

4.1.2 Coupled Trust-State Models

Following the development in the previous section, one can now define a more general trust-based interaction model that couples the trust evolution to how well adjacent agents are responding to an agent's movements. It is important to note that we call each player "agent", whether it be a human or a robot. By specifying the desired network performance through a more general, pairwise, symmetric, inter-agent performance cost $F_{ij}(\|x_i - x_j\|)$, as is done, for example, in the formation control literature, e.g., [54–56], the corresponding contribution by agent j to the increase in cost is given by

$$\frac{\partial F_{ij}(\|x_i - x_j\|)^T}{\partial x_j}\dot{x}_j,$$

which, in turn, should be coupled to the trust evolution.

Now, there are different ways in which this expression can be coupled to a trust model. For example, human trust is typically pairwise, i.e., a person does not necessarily trust all people equally. As such, a study of human-to-human interactions must capture this pairwise relationship. However, what makes human-to-autonomous-agent interactions different is that the trust is more uniform, i.e., a person may or may not trust the autonomous agents but will not necessarily be able to tell agents apart or form pairwise opinions about the performance of the agents. As such, we need two different types of models that reflect

these two different types of trusts.

4.1.2.1 Pairwise Trust

Consider a collection of N agents, interacting over a static, undirected, and connected information-exchange network, $G = (V, E)$. Here, the vertex set $V = \{1, \ldots, N\}$ is the set of agents, while the edge set $E \subset V \times V$ is a set of unordered pairs that encode the adjacency relationship in the network. Each agent has a state $x_i \in \mathbb{R}^d,\ i = 1, \ldots, N$, associated with it, representing, for example, a position in space or an opinion on a issue. Moreover, we add an additional state τ_{ij} to each ordered agent-pair in the network, which denotes agent i's level of trust for an adjacent agent j.

In light of the previous discussion regarding the two-agent case, given a pairwise performance cost $F_{ij}(\|x_i - x_j\|)$, we let the evolution of τ_{ij} depend on how much agent j's movement makes the performance cost decrease, i.e.,

Self-Centered Trust Model: Pairwise

$$\dot{\tau}_{ij} = -\frac{\partial F_{ij}(\|x_i - x_j\|)^T}{\partial x_j}\dot{x}_j,$$

or, in the case of the rendezvous problem, with

$$F_{ij}(\|x_i - x_j\|) = \frac{1}{2}\|x_i - x_j\|^2, \tag{4}$$

we get

$$\dot{\tau}_{ij} = (x_i - x_j)^T \dot{x}_j.$$

4.1.2.2 Neighborhood Trust

Rather than having the trust be a pairwise property, i.e., how much agent i trusts agent j, it could instead be a neighborhood property, i.e., how much agent i trusts its neighbors. The reason for this interpretation of trust is that the number of states could potentially grow very large as the network grows if trust was a pair-wise property. Moreover, in a network of largely anonymous agents, pair-wise relationships are, as already discussed, not a realistic feature.

Following the previous construction, we can let τ_i denote agent i's trust level, and simply use an aggregated update law

Self-Centered Trust Model: Neighborhood

$$\dot{\tau}_i = -\sum_{j \in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)^T}{\partial x_j} \dot{x}_j,$$

where N_i is the set of agents adjacent to agent i in the network., i.e., $N_i = \{j \in V \mid (i, j) \in E\}$. In the case of the rendezvous problem, this simplifies to

$$\dot{\tau}_i = \sum_{j \in N_i} (x_i - x_j)^T \dot{x}_j.$$

It should be noted here that these trust models are self-centered in that an agent only cares whether or not its neighbors are benefiting them directly. That is, agent i is only considering how much agent j is contributing to decreasing their pairwise cost F_{ij} in order to evaluate its trust in agent j. However, agent j could be helping to decrease some other cost in the network associated with it and an agent other than agent i and thus still contributing to the team, but agent i will not take this into account. We will handle this when we present the team-oriented trust model in Section 4.2.

4.1.2.3 Connecting Trust to State Evolution

These two trust models need to be coupled to the evolution of the physical states. Note that, in the absence of any trust states, a standard, gradient-descent-based update law is given by

$$\dot{x}_i = -\sum_{j \in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i},$$

which has been employed repeatedly in the literature for a number of different types of applications, including formation control, connectivity maintenance, and collision-avoidance, e.g., [54]. In this work, we augment this model by adding a trust gain to the evolution, i.e.,

$$\dot{x}_i = -\sum_{j \in N_i} \tau_{ij} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i},$$

or

$$\dot{x}_i = -\tau_i \sum_{j \in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i},$$

depending on which trust model is used. For the rendezvous problem, these system equations become

$$\dot{x}_i = \sum_{j \in N_i} \tau_{ij}(x_j - x_i), \tag{5}$$

or

$$\dot{x}_i = \tau_i \sum_{j \in N_i} (x_j - x_i). \tag{6}$$

4.1.2.4 Comparisons with Other Models

It should be noted that the dynamical models in (5) and (6) have several similarities to models used in the literature to describe the evolution of opinions in social groups. For example, DeGroot's model [57], given by

$$x(k+1) = \boldsymbol{W}x(k), \quad k = 0, 1, \ldots,$$

is a discrete-time linear model where $x(0) \in \mathbb{R}^n$ represents the initial opinions of n people and $\boldsymbol{W}$ is a static matrix representing how people's opinions influence each other. A similar model was introduced by Friedkin [58], given by

$$\boldsymbol{X}(k+1) = \boldsymbol{A}\boldsymbol{W}\boldsymbol{X}(k) + (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{X}(0)$$

where $\boldsymbol{W}$ is a static nonnegative matrix with row sums equal to one, which weights the influences from the other individuals and $\boldsymbol{A}$ is a static diagonal matrix, with $0 \leq a_{ii} \leq 1$ representing the amount that individual i is attached to his or her initial opinion.

Both of these models illustrate that an individual's opinion changes according to a weighted average of his or her own opinion and the opinions of his or her neighbors. Our model in (5) differs from these two models in that the influence weights are not static – they may change over time representing the fact that trust evolves over time. However, there are several existing models that incorporate weights that change over time.

For example, models have been introduced that account for biased assimilation, in which people bias the information obtained by their peers according to how similar it is to their own opinion [59]. Models such as that in [60] and [61] include this idea by modifying the weights in $\boldsymbol{W}$ according to how similar one's opinions are. In [61], weights can be either positive or negative with negative weights representing the fact that people may have negative ties to others that are dissimilar to them. Similarly, Hegselmann and Krause introduced a bounded-confidence model [29] where agents only interact if their difference in opinion is less than some threshold, d_0. This is equivalent to changing (5) by setting $\tau_{ij}(t) = 1$ if $\|x_i(t) - x_j(t)\| < d_0$ and $\tau_{ij}(t) = 0$ otherwise.

While these existing models incorporate time-varying influence weights, ours is different in that the weights (or trust values) do not depend on the difference in opinions of agents at a particular time instant but rather on the behavior or performance of an agent. This is because we are not modeling human opinion formation – we are representing interactions between both humans and robots, and it has been shown that a human's perceived trust in human-robot interaction is highly influenced by the robot's performance [18].

4.1.3 Invariance Results

One common feature of all the trust model variations previously discussed is that the total trust in the network is intimately linked to the performance of the system through an invariance, as already shown for the two-agent case. We first present such a result for the rendezvous problem under the collective trust model and then we generalize it to the case for which the network performance is evaluated through a desired performance cost.

4.1.3.1 Consensus With Collective Trust

Consider a system composed of N agents solving the rendezvous problem under the collective trust model:

$$\begin{aligned} \dot{x}_i &= \tau_i \sum_{j \in N_i} (x_j - x_i) \\ \dot{\tau}_i &= \sum_{j \in N_i} \left[\tau_j (x_j - x_i)^T \left(\sum_{k \in N_j} (x_j - x_k) \right) \right], \end{aligned} \tag{7}$$

where we have substituted the actual expressions for $\dot{x}_j$ in the trust model. In addition, let us assume that the individual states are scalars, and let us set $x = [x_1, \ldots, x_N]^T$ and $\tau = [\tau_1, \ldots, \tau_N]^T$.

The following invariance result holds for the rendezvous problem under the collective trust model as in (7).

Lemma 2. *Consider a collection of N agents under the dynamics in* (7)*, with $x_i \in \mathbb{R}$, then the following holds*

$$\frac{d}{dt}\left(\frac{1}{2}\|D^T x\|^2 + \mathbf{1}^T\tau\right) = 0.$$

with D the incidence matrix obtained by associating an arbitrary orientation with the network topology.

Proof. Let us define the total trust $\hat{\tau}$ in the network as

$$\hat{\tau} = \mathbf{1}^T\tau = \sum_{i=1}^{N} \tau_i,$$

and let us notice that the evolution of the total trust $\hat{\tau}$ is

$$\dot{\hat{\tau}} = \sum_{i=1}^{N}\left\{\sum_{j\in N_i}\left[\tau_j(x_j - x_i)^T\left(\sum_{k\in N_j}(x_j - x_k)\right)\right]\right\},$$

which, by rearranging the summation order (since the network is undirected), can be re-written as

$$\begin{aligned}\dot{\hat{\tau}} &= \sum_{j=1}^{N}\tau_j\left[\sum_{i\in N_j}(x_j - x_i)^T\left(\sum_{k\in N_j}(x_j - x_k)\right)\right]\\ &= \sum_{j=1}^{N}\left[\tau_j\left\|\sum_{i\in N_j}(x_j - x_i)\right\|^2\right].\end{aligned}$$

Now, letting L be the Laplacian associated with the information-exchange network, we have that

$$\mathbf{1}^T\dot{\tau} = x^T L\mathcal{T}Lx,$$

where $\mathbf{1} = [1, \ldots, 1]^T$ and $\mathcal{T} = \mathrm{diag}(\tau)$. We moreover observe that the x-dynamics becomes

$$\dot{x} = -\mathcal{T}Lx.$$

Furthermore, note that by associating an arbitrary orientation with the network, we can define the corresponding incidence matrix D, and

$$\frac{1}{2}\frac{d}{dt}\|D^T x\|^2 = \frac{1}{2}\frac{d}{dt}(x^T DD^T x) = x^T L\dot{x} = -x^T L\mathcal{T} Lx,$$

which is exactly equal to $-\mathbf{1}^T\dot{\tau}$, and thus the result follows. ■

We point out that this invariance result is directly analogous to the two-agent invariance result given in Lemma 1, and what it tells us is that an overall increase in performance must correspond to a similar increase in the total trust in the network. Or, negatively, that if the total trust is reduced then the network performance has to be reduced as well.

Notably, it turns out that this holds true also in the more general cases (without cut-offs), whereby the network performance is evaluated through the performance cost

$$F(x) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i} F_{ij}(\|x_i - x_j\|),$$

as discussed in the following two subsections.

4.1.3.2 The General Neighborhood Trust Case

Let us consider the neighborhood-based trust model, for which the coupled dynamics of an agent i is

$$\begin{aligned}\dot{x}_i &= -\tau_i \sum_{j\in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i},\\ \dot{\tau}_i &= -\sum_{j\in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)^T}{\partial x_j}\dot{x}_j.\end{aligned} \tag{8}$$

And by letting $x = [x_1^T, \ldots, x_N^T]^T$ and $\tau = [\tau_1, \ldots, \tau_N]^T$, the following invariance result holds.

Theorem 2. *Consider a collection of N agents under the dynamics in* (8)*, with $x_i \in \mathbb{R}$, then the following holds*

$$\frac{d}{dt}(F(x) + \mathbf{1}^T\tau) = 0.$$

Proof. To prove this result, let us notice that the derivative of the performance cost $F(x(t))$ is given by

$$\frac{d}{dt}F(x(t)) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i}\left(\frac{\partial F_{ij}}{\partial x_i}\dot{x}_i + \frac{\partial F_{ij}}{\partial x_j}\dot{x}_j\right),$$

which, since the network is undirected and the performance costs are symmetric, i.e. $F_{ij} = F_{ji}$, simplifies to

$$\frac{d}{dt}F(x(t)) = \sum_{i=1}^{N}\sum_{j\in N_i}\frac{\partial F_{ij}}{\partial x_j}\dot{x}_j.$$

Similarly, the total trust evolution, by definition, is given by

$$\dot{\hat{\tau}} = \sum_{i=1}^{N}\dot{\tau}_i = -\sum_{i=1}^{N}\sum_{j\in N_i}\frac{\partial F_{ij}}{\partial x_j}\dot{x}_j.$$

Therefore, the result follows. ∎

4.1.3.3 The General Pairwise Trust Case

Let us consider the pairwise-based trust model, for which the coupled dynamics of an agent i is

$$\begin{aligned}\dot{x}_i &= -\sum_{j\in N_i}\tau_{ij}\frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i},\\ \dot{\tau}_{ij} &= -\frac{\partial F_{ij}(\|x_i - x_j\|)^T}{\partial x_j}\dot{x}_j,\end{aligned} \tag{9}$$

and for which the total trust in the network is redefined as

$$\hat{\tau} = \sum_{i=1}^{N}\sum_{j\in N_i}\tau_{ij}.$$

Then, letting $x = [x_1^T, \ldots, x_N^T]^T$, the following invariance result holds.

Theorem 3. *Consider a collection of N agents under the dynamics in* (9)*, with $x_i \in \mathbb{R}$, then the following holds*

$$\frac{d}{dt}(F(x) + \hat{\tau}) = 0.$$

Proof. To prove this result, let us notice that the total trust evolution is given by

$$\dot{\hat{\tau}} = \sum_{i=1}^{N}\sum_{j\in N_i}\dot{\tau}_{ij} = -\sum_{i=1}^{N}\sum_{j\in N_i}\frac{\partial F_{ij}}{\partial x_j}\dot{x}_j$$

and let us recall from the proof of Theorem 2 that

$$\frac{d}{dt}F(x(t)) = \sum_{i=1}^{N}\sum_{j\in N_i}\frac{\partial F_{ij}}{\partial x_j}\dot{x}_j.$$

At this point, by combining these two results the result follows. ■

As a final note, we can stack the pairwise trusts to form an aggregated trust vector for each agent, where $\tau_i \in \mathbb{R}^{|N_i|}$, and let $\tau_i = [\tau_{ij_1}, \ldots, \tau_{ij_{|N_i|}}]^T$, where $N_i = \{j_1, \ldots, j_{|N_i|}\}$. Then we stack the individual agents' aggregated trust vectors to get the overall aggregated pairwise trust vector, $\tau = [\tau_1^T, \ldots, \tau_N^T]^T$, allowing us to write the invariance in exactly the same way as we did in the neighborhood trust case,

$$\frac{d}{dt}(F(x) + \mathbf{1}^T\tau) = 0.$$

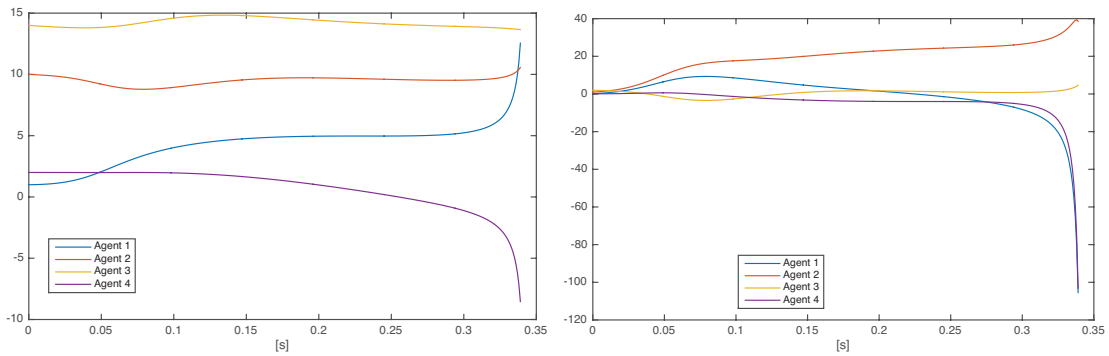
4.1.3.4 Interpretations

These invariance results are directly analogous to the two-agent invariance result given in Lemma 1, and indicate that an overall increase in performance must correspond to a similar increase in the total trust in the network. Or, negatively, that if the total trust is reduced then the network performance has to be reduced as well. As already pointed out for the two-agent case, this result aligns well with the studies from organizational psychology that show a strong positive relation between trust and performance in teams within organizations [62]. Furthermore, if there is not sufficient trust between agents, the opinions may diverge, exhibiting belief polarization. And, as seen in the two-agent case, this might even happen in finite time, since the coupled dynamics is not globally Lipschitz.

Furthermore, as it can be expected, the general case of N-agents, where $N > 2$, is substantially more complex than the case of two agents. For example, a result which relates the achievement of a desired collective behavior to the initial trust of the agents, as given in Theorem 1 for the two-agents case, no longer exists as demonstrated by the following numerical example.

Consider a system composed of four agents for which the interactions are dictated by the line topology depicted in Figure 4. Assume that the agents are to solve the rendezvous

Figure 4. Line-based graph topology



(a) Agents' collective states evolution. (b) Agents' trust states evolution

Figure 5. Simulation involving four agents interacting according to a line-based graph topology. It can be noticed how a finite-escape time occurs at time 0.35**s.**

problem under the collective trust model as in (7) and consider the following set of initial conditions

$$x(0) = [1\ 10\ 14\ 2]^T, \quad \tau(0) = [.1\ 1\ 2\ .1]^T. \tag{10}$$

Figures 5a and 5b depict the collective dynamics and the trust dynamics over time, respectively. It can be noticed that a finite-escape time occurs around $t = .35s$ even though the initial trusts of the agents are positive and so is their sum. This clearly violates the claim of Theorem 1, and thus demonstrates that the problem becomes significantly more involved for the N-agent case.

Intuitively, this phenomenon can be explained by the fact that Theorem 1 only provides a result concerning the evolution of the sum of the agents' trusts. Notably, for the consensus dynamics, in the case of two-agents, this also suffices to constrain the evolution of the two agents' trusts. Unfortunately, this bind between the evolution of the sum and the evolution of the agents' trusts no longer exists for the N-agent case, and we cannot prevent a finite-escape time to occur simply by looking at the initial value of the sum of the trusts.

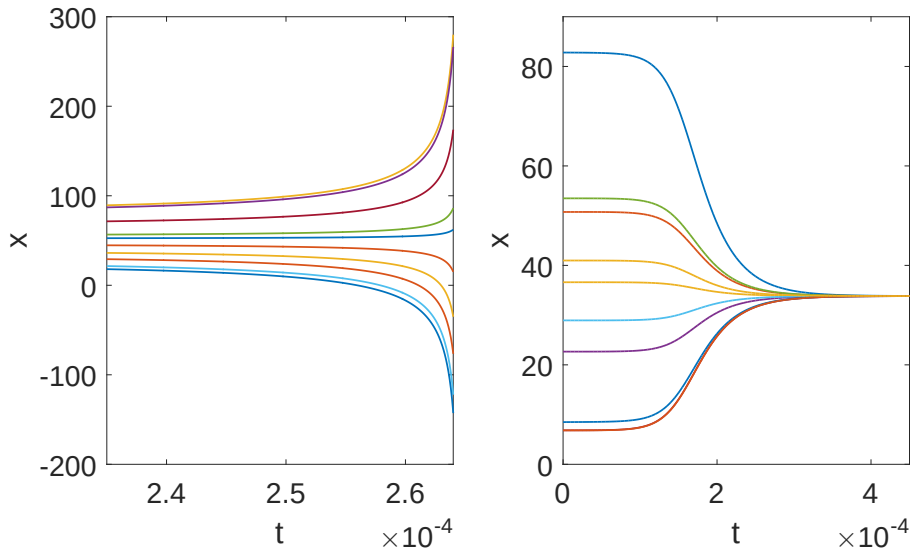


Figure 6. State trajectories for the 10-agent system with trust are shown. In the figure on the left, the states diverge and in the figure on the right, the states converge to a common value.

4.1.4 Simulations

While the conditions on the initial trust values required for convergence and divergence are clear in the two-agent case, they do not extend to the general N-agent case. Even if the sum of the initial trust values is positive, the states may still diverge, as shown in the left plot in Figure 6. Here, 10 agents are simulated for a complete network using the the rendezvous cost in (4) and the dynamics in (8). For this example, the initial states values were chosen from the interval $(0, 100)$ and the sum of the initial trust values is 6.7849. Even though this sum is positive, the states still diverge.

The reverse is true as well; even if the sum of the initial trust values is negative, the states may actually still converge. The plot on the right-hand side of Figure 6 depicts a simulation of 10 agents, also executing rendezvous using the dynamics in (8) over a complete network. In this example, the sum of the initial trust values is -3.7820, but the system still achieves convergence. Hence, for the general N-agent case, the convergence conditions not only depend on the sum of the initial trust values but also on the initial states and on the topology of the network.

4.1.4.1 *Belief Polarization*

In order to capture the belief polarization phenomenon, then the update laws simply have to be adjusted to ensure that only neighboring agents that are sufficiently trusted are taken into account. If we let $\bar{\tau}$ denote this threshold, the update equations for the neighborhood trust scenario become

$$\dot{x}_i = \begin{cases} -\tau_i \sum_{j \in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i} & \text{if } \tau_i \geq \bar{\tau} \\ 0 & \text{otherwise.} \end{cases}$$

For pairwise trust, we redefine the neighborhood as

$$N_i(\tau) = \{j \in V \mid (i, j) \in E \text{ and } \tau_{ij} \geq \bar{\tau}\},$$

with the update law becoming

$$\dot{x}_i = -\sum_{j \in N_i(\tau)} \tau_{ij} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i}. \tag{11}$$

Examples of running (11) over 5 agents are shown in Figure 7, with random initial conditions over x (uniform over $[0, 1]$) and τ (normally distributed with zero mean), with $\bar{\tau} = 0.3$. As shown, depending on the initial states and trusts, dramatically different results are obtained.

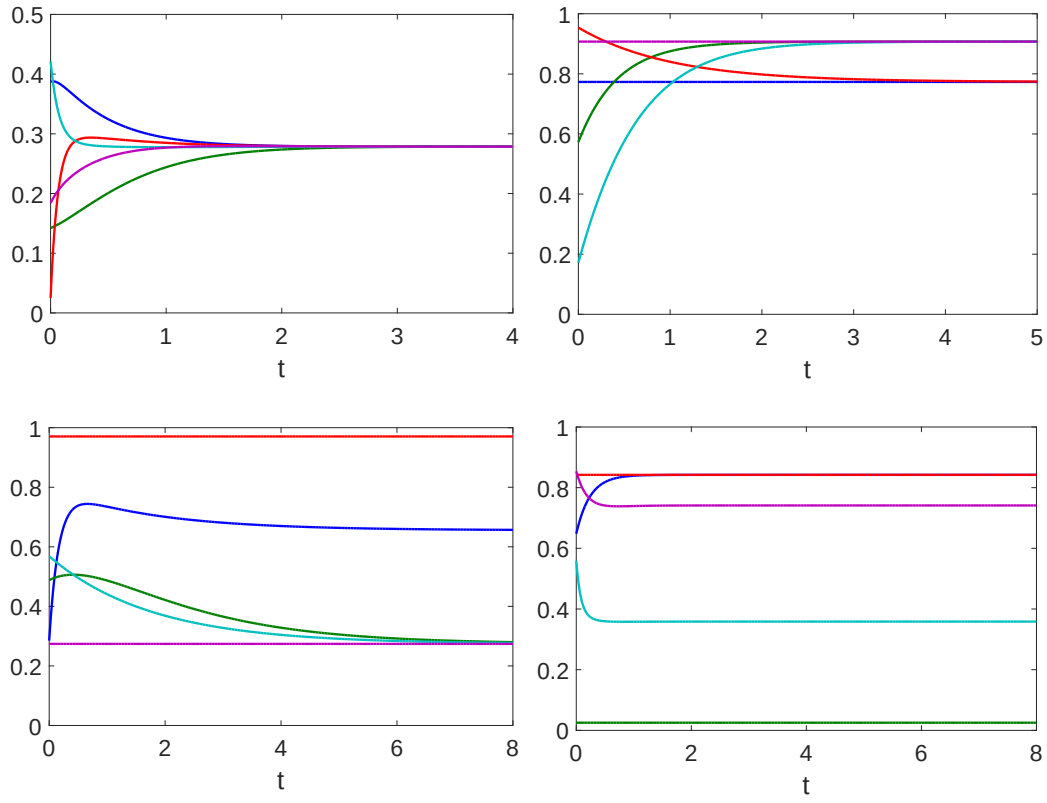


Figure 7. Four different outcomes are shown, where five agents cluster into one, two, three, and even four different groups, respectively, as a result of the random initial conditions on the state and trust values.

4.1.5 Comparisons: With Trust vs. Without Trust

Now that we know under what conditions the two-agent system results in convergence of the agents' states, we return to the two-agent system and compare our proposed dynamics to the classical consensus dynamics in the literature, e.g., [5]. In particular, we compare rates of convergence and agreement states in order to determine if the addition of trust helps the states converge any faster and how trust affects the point at which the agents meet.

4.1.5.1 Convergence Rates

We aim to compare the two-agent scalar system with trust-based weights, given in (3), to the two-agent scalar system without trust, given by

$$\begin{aligned} \dot{x}_1 &= \alpha_1(x_2 - x_1) \\ \dot{x}_2 &= \alpha_2(x_1 - x_2). \end{aligned} \tag{12}$$

where $\alpha_1 \in \mathbb{R}$ and $\alpha_2 \in \mathbb{R}$ are static weights. We will compare how quickly each of the systems converges and also the points at which each of the systems converge. The first question that we consider is whether agreement is achieved faster when trust plays a role in the decision making process as compared to when trust is not taken into consideration.

In order to answer this question, we analyze the systems using a disagreement vector, δ, where

$$\delta(t) = \frac{1}{2}(x_1(t) - x_2(t))^2.$$

And, to ensure convergence for the two-agent case, we must have that $\hat{\tau}(0) > 0$, where $\hat{\tau}(0) = \tau_1(0) + \tau_2(0)$ which is assumed to hold true for the remainder of this section.

Under the dynamics without trust given in (12), the time derivative of the nominal disagreement, δ_N, is

$$\dot{\delta}_N = -2(\alpha_1 + \alpha_2)\delta_N$$

which is a linear time-invariant system with solution

$$\delta_N(t) = e^{-2(\alpha_1+\alpha_2)t}\delta_N(0). \tag{13}$$

Similarly, we can analyze the system with trust, under the dynamics in (3), by taking the time derivative of the disagreement, δ_T, to get

$$\dot{\delta}_T = 2\delta_T(\delta_T - C), \tag{14}$$

and it is thus straightforward to show the following result.

Lemma 3. *The disagreement under the dynamics in* (3)*, is given by*

$$\delta_T(t) = \frac{C}{1 + \frac{\hat{\tau}(0)}{\delta(0)} e^{2Ct}}, \tag{15}$$

where

$$C = \hat{\tau}(0) + \delta_T(0).$$

We can now compare the disagreement under the system with trust, δ_T, to the disagreement under the system without trust, δ_N, in order to determine which system converges faster. This result is shown in the following theorem.

Theorem 4. *If* $\hat{\tau}(0) \geq \alpha_1 + \alpha_2$ *and* $\delta_N(0) = \delta_T(0)$*, then* $\delta_T(t) \leq \delta_N(t)\ \forall t > 0$.

Proof. By noting that C is invariant, we can substitute $C = \hat{\tau}(t) + \delta_T(t)$ into (14) to get

$$\dot{\delta}_T(t) = -2\hat{\tau}(t)\delta_T(t)$$

and, because $\hat{\tau}(0) > 0$, we know that $\hat{\tau}$ will increase monotonically until $\delta_T = 0$ in the limit. Combining this with the two-agent invariance result in Lemma 1, we get

$$\hat{\tau}(0) \leq \hat{\tau}(t) \leq \hat{\tau}(0) + \delta_T(0) = C.$$

Therefore, we have that

$$-2C\delta_T(t) \leq \dot{\delta}_T(t) \leq -2\hat{\tau}(0)\delta_T(t)$$

and thus, using Grönwall's inequality, we have

$$e^{-2Ct}\delta_T(0) \leq \delta_T(t) \leq e^{-2\hat{\tau}(0)t}\delta_T(0).$$

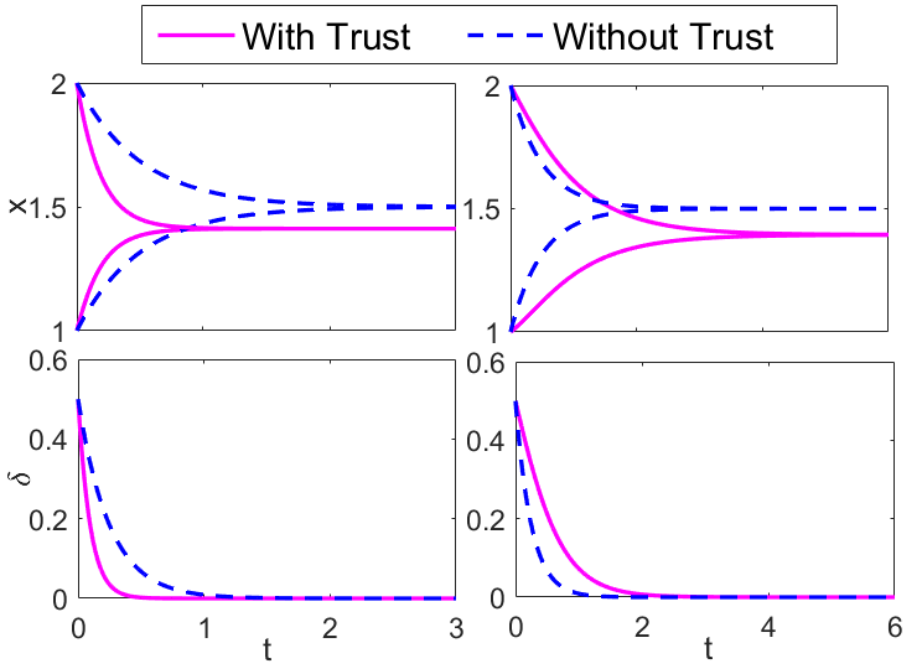


Figure 8. The state trajectories in the upper left corner indicate when $\tau_1(0) = 2$ and $\tau_2(0) = 3$ and thus the system with trust (solid curves) converges faster than the one without trust (dashed curves). The trajectories in the upper right corner were created using $\tau_1(0) = 0.2$ and $\tau_2(0) = 0.5$ and the system without trust converges faster than that with trust. The bottom two figures show the disagreement, δ, for both cases.

Taking the right hand side of this inequality and combining it with the fact that $\hat{\tau}(0) \geq \alpha_1 + \alpha_2$, yields the desired result,

$$\delta_T(t) \leq e^{-2\hat{\tau}(0)t}\delta_T(0) \leq e^{-2(\alpha_1+\alpha_2)t}\delta_T(0) = \delta_N(t).$$

■

It can be concluded that the system with trust converges faster than that without trust, provided that $\hat{\tau}(0) > \alpha_1 + \alpha_2$. This result is illustrated in Figure 8, where the curves in the plots on the left-hand side for the system with trust have $\tau_1(0) = 2$ and $\tau_2(0) = 3$, and $\alpha_1 = \alpha_2 = 1$, giving $\hat{\tau}(0) > \alpha_1 + \alpha_2$ and thus the system with trust converges faster than that without trust. In the plots on the right-hand side, the system with trust was initialized using $\tau_1(0) = 0.2$ and $\tau_2(0) = 0.5$, and again $\alpha_1 = \alpha_2 = 1$, yielding a sum $\hat{\tau}(0) < \alpha_1 + \alpha_2$ and thus the system with trust is not guaranteed to converge faster than the system without trust and indeed it does not. In both scenarios, $x_1(0) = 1$ and $x_2(0) = 2$.

As a result of Theorem 4, the agents will agree faster under the trust-based dynamics than under the standard consensus protocol provided there is sufficient initial trust in the system. And, relating this to psychology, we can think of this intuitively as the existence of trust speeding up decision making in teams.

4.1.5.2 Agreement States

In addition to comparing the convergence rates of these systems, we are also interested in determining where the agents converge, or in other words, what point they agree upon. This will provide intuition into how trust affects the decisions made by teams. We call this point an *agreement state*, represented by $\bar{x}$, where, in the two-agent case,

$$\bar{x} = \lim_{t\to\infty} x_1(t) = \lim_{t\to\infty} x_2(t),$$

provided that the states do indeed converge. In order to compare agreement states between the systems with trust and without, we need to find the agreement state for the system with trust, which is shown in the following theorem. For notational simplicity, we let $\tau_{i0} = \tau_i(0)$ and $x_{i0} = x_i(0)$, for $i \in \{1, 2\}$.

Theorem 5. *Under the two-agent dynamics in* (3)*, with* $\tau_{10} + \tau_{20} > 0$*, the following holds: If* $x_{10} \neq x_{20}$*, then*

$$\bar{x} = \frac{(\tau_{10} - \tau_{20})}{4(x_{10} - x_{20})(D+1)^{3/2}} \left[D \ln\left(\frac{D}{2 + D + 2\sqrt{D+1}} \right) - 2\sqrt{D+1} \right] + \frac{x_{10} + x_{20}}{2} \tag{16}$$

where

$$D = \frac{\hat{\tau}(0)}{\delta(0)} = \frac{2(\tau_{10} + \tau_{20})}{(x_{10} - x_{20})^2}$$

and, if $x_{10} = x_{20}$*, then the agents have already trivially reached agreement and thus*

$$\bar{x} = x_{10} = x_{20}.$$

Proof. We can find where the agents converge by considering the aggregate system

$$\begin{bmatrix} \hat{\tau} \\ \tau' \\ \hat{x} \\ x' \end{bmatrix} = \begin{bmatrix} \tau_1 + \tau_2 \\ \tau_1 - \tau_2 \\ x_1 + x_2 \\ x_1 - x_2 \end{bmatrix}.$$

First, (15) can be rewritten to get

$$(x_1(t) - x_2(t))^2 = \frac{2C}{1 + De^{2Ct}} \tag{17}$$

and we solve for $\hat{\tau} = \tau_1 + \tau_2$ by noting that

$$\dot{\hat{\tau}} = \dot{\tau}_1 + \dot{\tau}_2 = (x_1(t) - x_2(t))^2 \, \hat{\tau}.$$

This is a linear time-varying system which can be solved for by integrating $(x_1(t) - x_2(t))^2$, which we know from (17) and the solution is

$$\hat{\tau}(t) = \tau_1(t) + \tau_2(t) = \frac{(D+1)(\tau_1(0) + \tau_2(0))}{D + e^{-2Ct}}. \tag{18}$$

A similar procedure is used to find $\tau' = \tau_1 - \tau_2$, resulting in

$$\tau'(t) = \tau_1(t) - \tau_2(t) = \frac{(D + e^{-2Ct})(\tau_1(0) - \tau_2(0))}{D+1}.$$

Now that we have closed-form expressions for $\hat{\tau}(t)$ and $\tau'(t)$, we solve for $x' = x_1 - x_2$ by noting that

$$\dot{x}' = \dot{x}_1 - \dot{x}_2 = -\hat{\tau} x',$$

which is also a linear time-varying system which can be solved using (18) to get

$$x'(t) = x_1(t) - x_2(t) = (x_1(0) - x_2(0)) \sqrt{\frac{1+D}{1 + De^{2Ct}}}.$$

Lastly, we solve for $\hat{x} = x_1 + x_2$ by noting that

$$\dot{\hat{x}} = \dot{x}_1 + \dot{x}_2 = -(\tau_1 - \tau_2)(x_1 - x_2) = -\tau' \, x'$$

where we have closed-form expressions for both $\tau'(t)$ and $x'(t)$, so the entire right-hand side is known. Thus we integrate to obtain the solution

$$\hat{x}(t) = \frac{-(\tau_{10} - \tau_{20})(x_{10} - x_{20})}{\sqrt{D+1}} [j(t) - j(0)] + \hat{x}(0), \tag{19}$$

where

$$j(t) = \frac{-1}{4C}\left[2e^{-2Ct}\sqrt{1 + De^{2Ct}} + D\ln\left(e^{-2Ct}\left(2 + De^{2Ct} + 2\sqrt{1 + De^{2Ct}}\right)\right)\right]. \tag{20}$$

and where we let $\tau_{i0} = \tau_i(0)$ and $x_{i0} = x_i(0)$, for $i = 1, 2$.

In order to find the meeting point, $\bar{x}$, we need $\lim_{t\to\infty} x_1(t)$ (or alternatively $\lim_{t\to\infty} x_2(t)$). Noting that

$$x_1(t) = \frac{1}{2}(x_1(t) + x_2(t)) + \frac{1}{2}(x_1(t) - x_2(t)) = \frac{1}{2}(\hat{x}(t) + x'(t)),$$

and, as long as $\tau_1(0) + \tau_2(0) > 0$, then $x'(t) \to 0$ as $t \to \infty$, which allows the meeting point to be expressed as

$$\bar{x} = \lim_{t\to\infty} x_1(t) = \lim_{t\to\infty} x_2(t) = \frac{1}{2}\lim_{t\to\infty} \hat{x}(t).$$

Thus, by substituting in $\hat{x}(t)$ from (19) and (20), we get

$$\bar{x} = \frac{(\tau_{10} - \tau_{20})(x_{10} - x_{20})}{8C\sqrt{D+1}}\left[D\ln\left(\frac{D}{2 + D + 2\sqrt{D+1}}\right) - 2\sqrt{D+1}\right] + \frac{\hat{x}(0)}{2},$$

and, by noting that $C = \delta(0)(D+1)$, we get

$$\bar{x} = \frac{(\tau_{10} - \tau_{20})}{4(x_{10} - x_{20})(D+1)^{3/2}}\left[D\ln\left(\frac{D}{2 + D + 2\sqrt{D+1}}\right) - 2\sqrt{D+1}\right] + \frac{\hat{x}(0)}{2}.$$

■

Immediately, it can be noted that when $\tau_{10} = \tau_{20}$, the meeting point is the same as that of standard consensus, the average of the initial states of the agents, $\hat{x}(0)/2$. We can also show that when $\tau_{10} > \tau_{20}$, the agents meet closer to agent 2's initial state and when $\tau_{20} > \tau_{10}$, the agents meet closer to agent 1's intial state.

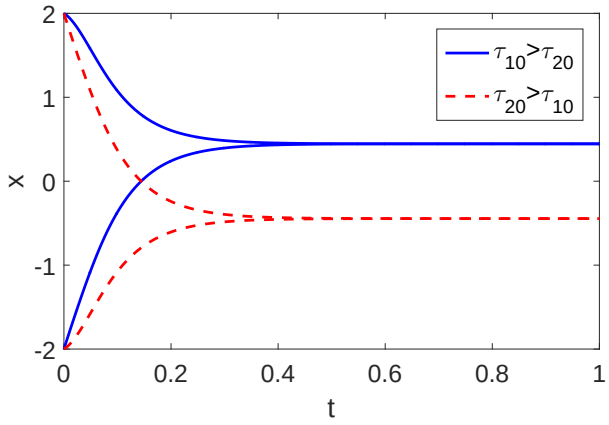


Figure 9. Two sets of state trajectories for the two-agent system with trust are shown, where each converge to a different meeting point depending on the initial trust values.

If we have that $\tau_{10} > \tau_{20}$ and $x_{10} > x_{20}$, then

$$\bar{x} < \frac{x_{10} + x_{20}}{2}$$

and thus the agents meet closer to the initial position of agent 2 than that of agent 1. This seems to make sense intuitively; if I trust my friend more than my friend trusts me, it is plausible that we will decide on an option that is closer to my friend's initial opinion than my own. Similar analysis can be done for the case when $\tau_{20} > \tau_{10}$, resulting in the agents meeting closer to the initial position of agent 1.

Figure 9 illustrates this result in simulation. If we let $x_{10} = -2$ and $x_{20} = 2$, and first let $\tau_{10} = 5$ and $\tau_{20} = 1$, then the agents meet at $\bar{x} = 0.4455$, which is closer to the initial state of agent 2. If we swap the initial trust values so that $\tau_{10} = 1$ and $\tau_{20} = 5$, and keep the initial x-values the same, then the agents meet at $\bar{x} = -0.4455$, which is closer to the initial state of agent 1, illustrating the previous result that the agents meet closer to the agent with the lower initial trust.

Furthermore, we can show that two agents may even meet outside the convex hull of their initial positions, which is consistent with the notion of group polarization [63], where members of a group arrive at decisions that are more extreme than the opinions of the individuals before the commencement of the group discussion. And, by exhibiting this

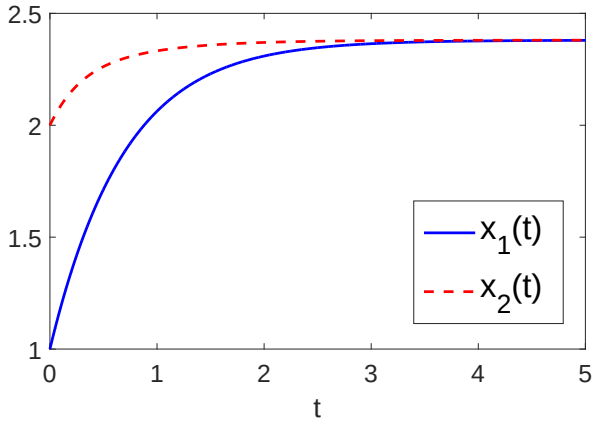


Figure 10. State trajectories for the two-agent system with trust are shown, in which the agents converge to a point that is outside of the convex hull of their initial states.

behavior, our trust notion satisfies the remaining criteria needed to be considered psychologically consistent. As seen in Figure 10, two agents with initial positions and trust values of $x_{10} = 1$, $x_{20} = 2$, $\tau_{10} = 2$, and $\tau_{20} = -1$, end up meeting at $\bar{x} = 2.3802$, which is outside of the convex hull of the initial positions of the agents. We can also note that this is indeed closer to the initial position of agent 2, since we have that $\tau_{10} > \tau_{20}$.

4.1.6 Heterogeneous Dynamics: 1 Human, N-1 Robots

As alluded to earlier, one of the objectives of this thesis is to analyze the behavior of human-robot teams. While humans operate under some notion of trust, it may not always be necessary for robots to keep track of trust as they are designed to produce the expected behavior and do not "feel" in the same way that humans do. In this section we focus on the case where there is one human, with trust dynamics, and $N - 1$ robots, without trust dynamics. We examine this particular case in the context of the rendezvous problem.

Consider a collection of N agents, where the human agent is denoted by $i = 1$ and the robots are denoted $i \in \{2, ..., N\}$. We let the static, undirected information-exchange network be given by $G = (V, E)$ and let the state of agent i, $x_i \in \mathbb{R}$, be scalar for the sake of notational simplicity. Only the human keeps track of a trust state, $\tau \in \mathbb{R}$. For the human,

$$\dot{x}_1 = \tau \sum_{j \in N_1} (x_j - x_1) \tag{21}$$

and for the robot agents, that is $i \in \{2, ..., N\}$,

$$\dot{x}_i = \sum_{j \in N_i} (x_j - x_i). \tag{22}$$

Trust evolution is given by the neighborhood trust dynamics discussed previously,

$$\dot{\tau} = \sum_{j \in N_1} (x_1 - x_j)\dot{x}_j = \sum_{j \in N_1} \left((x_1 - x_j) \sum_{k \in N_j} (x_k - x_j) \right). \tag{23}$$

If we let $x = [x_1, x_2, ..., x_N]^T$, and L be the graph Laplacian associated with G and define a function $B : \mathbb{R} \to \mathbb{R}^{N \times N}$ such that $B(\tau) = \text{diag}\{\tau, 1, ..., 1\}$, then the state dynamics can be written in a compact form as

$$\dot{x} = -B(\tau)Lx. \tag{24}$$

The first thing that we can note about this system is that as long as the information-exchange graph G is connected and all agents are neighbors of agent 1, then the trust will always be non-decreasing, as shown in the following Lemma.

Lemma 4. *Consider a collection of N-agents with state dynamics given in* (24) *and trust dynamics for the human given in* (23)*, where the underlying information-exchange graph* $G(V, E)$ *is connected and* $N_1 = \{2, 3, \ldots, N\}$*. Then* $\dot{\tau}(t) \geq 0$ *for all* $t \geq 0$*.*

Proof. Rearranging (23) gives

$$\dot{\tau} = \sum_{j \in N_1} \sum_{k \in N_j} (x_1 - x_j)(x_k - x_j).$$

If we fix a $j \in N_1$, then for every $k \in N_j$, we get a term a_{jk} in the summation such that $a_{jk} = (x_1 - x_j)(x_k - x_j)$. Then there are two separate cases: either $k = 1$ or $k \in N_j \setminus \{1\}$. If $k = 1$, then the term is $(x_1 - x_j)^2$, and in the other case, since $k \in N_1$ by assumption, we will also have the complement term $a_{kj} = (x_1 - x_k)(x_j - x_k)$. When we add the two terms, we get $a_{jk} + a_{kj} = (x_k - x_j)^2$. Hence,

$$\dot{\tau}(t) = \sum_{(j,k) \in E} (x_k(t) - x_j(t))^2 \geq 0. \tag{25}$$

■

Furthermore, unless all the agents are in agreement (that is, $x_k = x_j$ for all $(j,k) \in E$), then the summation in (25) is strictly positive, and we have the following immediate corollary.

Corollary 1. *Provided that the information exchange network $G(V,E)$ is connected and $N_1 = \{2,3,\ldots,N\}$, then $\dot{\tau}(t) = 0$ if and only if $x_k(t) = x_j(t)$ for all $k, j \in \{1,\ldots,N\}$.*

Given the previous results, we would like to show that the states of this system will converge to a common value, achieving consensus, regardless of the initial trust value, $\tau(0)$. To do so, we allow the trust τ to evolve until it becomes positive and then tackle the problem after that point. If we let t' denote the first time such that $\tau(t') > 0$, then we can conclude that $\tau(t) > 0$ for all $t > t'$. This idea is used in the following theorem to prove the convergence of the system.

Lemma 5. *If $\tau > 0$ and L is the graph Laplacian associated with a connected graph $G(V,E)$, then $L_\tau = B(\tau)L$ is positive semi-definite and the null space of L_τ is* span$\{\mathbb{1}\}$.

Proof. First, it can be shown that $L_\tau = B(\tau)L$ is similar to the matrix $S = B^{1/2}LB^{1/2}$ (where the dependence on τ was dropped for notational simplicity) by pre-multiplying L_τ by $B^{-1/2}$ and post-multiplying by $B^{1/2}$ to get S. We can rewrite S as $B^{1/2}DD^TB^{1/2}$, because L is symmetric and has non-negative eigenvalues due to the fact that $G(V,E)$ is connected. Therefore the eigenvalues of S are non-negative and thus the eigenvalues of L_τ are also non-negative because the two are similar. It is straightforward to show that the null space of $B(\tau)L$ is the same as the null space of L given that $B(\tau)$ is invertible as long as $\tau \neq 0$. And, it is known that the null space of L for a connected graph is span$\{\mathbb{1}\}$. ∎

Theorem 6. *Given a collection of N-agents with dynamics given in* (21) *and* (22) *and trust dynamics for the human given in* (23) *with a connected information-exchange graph $G(V,E)$ with $N_1 = \{2,3,\ldots,N\}$, all agents' states asymptotically converge to a common value. That is, if we let $x = [x_1, x_2, ..., x_N]^T$, then $x(t) \to \rho\mathbb{1}$ as $t \to \infty$ for some $\rho \in \mathbb{R}$.*

Proof. To analyze this system, we can form the Lyapunov function

$$V(x) = \frac{1}{2}x^T x = \frac{1}{2}\|x\|^2$$

with

$$\dot{V} = -x^T B(\tau)Lx,$$

where, because of Lemma 5, we know that $\dot{V} \leq 0$ provided that $\tau > 0$ and furthermore, $\dot{V} < 0$ as long as $x \notin \text{span}\{\mathbb{1}\}$.

Thus, we need $\tau(t') > 0$ for some $t' \geq 0$. If $\tau(0) > 0$, then this is satisfied trivially, and by Lemma 4, $\tau(t) \geq \tau(0) > 0$, for all $t > 0$. Thus, $B(\tau(t))L \geq 0$ for all $t > 0$ and therefore $\dot{V} < 0$ for all $t > 0$ and for all $x \notin \text{span}\{\mathbb{1}\}$. However, if $\tau(0) \leq 0$, then one of two things can happen: either there exists a $T \geq 0$ such that $\tau(T) = 0$ with $\dot{\tau}(T) > 0$ or $\lim_{t\to\infty} \tau(t) = -k^2$ where $k \geq 0$ is a constant.

In the latter case, however, $\lim_{t\to\infty} \dot{\tau}(t) = 0$, and by Corollary 1, this implies that $\lim_{t\to\infty} x(t) = \rho\mathbb{1}$, and thus consensus is achieved. In the former case, because there exists a $T \geq 0$ with $\tau(T) = 0$ and $\dot{\tau}(T) > 0$, there must also exist a $t' > T$ such that $\tau(t') > 0$ and by Lemma 4, $\tau(t) \geq \tau(t') > 0$, for all $t > t'$. Thus, $B(\tau(t))L \geq 0$ for all $t > t'$ and therefore $\dot{V} < 0$ for all $t > t'$ and for all $x \notin \text{span}\{\mathbb{1}\}$.

In the cases where $\dot{V} \leq 0$, we use LaSalle's invariance principle and find the largest invariant set contained in the set $\{x \in \mathbb{R}^N \mid \dot{V}(x) = 0\}$, which is $M = \text{span}\{\mathbb{1}\}$. Therefore, $x \to\in M$ as $t \to \infty$, or in other words, $\lim_{t\to\infty} x(t) = \rho\mathbb{1}$ for some $\rho \in \mathbb{R}$. ∎

This result aligns with intuition – all of the robot agents are designed to do the "right" thing, so the human's trust in those agents will increase, and therefore the human agent will eventually also do the "right" thing (when its trust value becomes positive), which is to evolve its state in such a way that it becomes closer to its neighbors' states. It is also important to point out that this restriction on the topology only affects the edges between the human and the robots, but does not restrict the edges between robots.

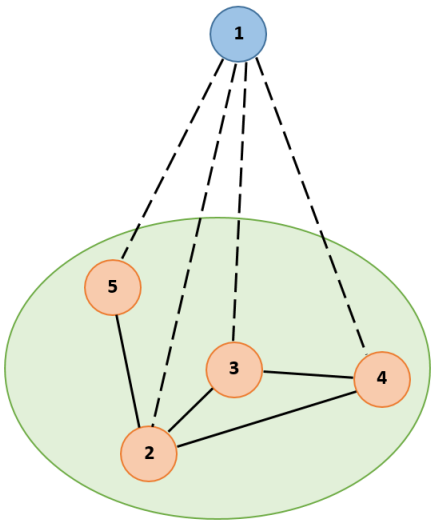


Figure 11. Depicted is an example of a network topology that converges under the heterogeneous dynamics protocol. The nodes contained in the oval represent the robot agents and the one outside the oval represents the human agent.

We can, in fact, view this as having two layers in the system – one layer consists only of the human agent and the second layer consists of all of the robot agents. If we let $G_r = (V_r, E_r)$ be the subgraph formed by only looking at the robot agents, with vertex set $V_r = \{2, \ldots, N\}$ and edge set $E_r = \{(i, j) \in E \,|\, i, j \in V_r\}$, then the only restriction for the robot layer required by Theorem 6 is is that G_r is connected. However, the human must interact with every robot to ensure that it is not being deceived, which will require an edge $(1, k) \in E$ to exist for all $k \in V_r$. An example of this kind of topology is shown in Figure 11, where the dashed lines represent the edges between the human agent and the robot agents and the solid lines represent the edges in E_r, which pertain only to the robot network subgraph G_r. Note that the subgraph G_r in this network is not complete because there are missing edges between nodes 3 and 5 and between nodes 4 and 5.

4.2 Team-Oriented Trust

In the previous section, we presented a trust model where agent i's trust in its neighbors evolves according to

$$\dot{\tau}_i = -\sum_{j \in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_j}^T \dot{x}_j, \tag{26}$$

where $F_{ij}(\|x_i - x_j\|)$ is the inter-agent performance cost that agents i and j collectively aim to minimize. This is a self-centered trust notion in that agent i only cares about whether its neighbors are helping to decrease the portions of the cost that are pertinent to agent i. However, an agent could be instantaneously contributing to decreasing the overall team cost, $F(x)$, but not necessarily contributing to decreasing agent i's portion of the cost at a particular point in time, and this trust model could potentially deem it untrustworthy.

For example, suppose the goal of the agents was to meet, or in other words achieve consensus, by minimizing the performance cost

$$F(x) = \sum_{(i,j) \in E} F_{ij}(\|x_i - x_j\|),$$

where

$$F_{ij}(\|x_i - x_j\|) = \frac{1}{2}\|x_i - x_j\|^2 \tag{27}$$

and suppose the initial configuration of the agents is as in Figure 12, where the lines between agents represent links indicating edges in the graph. Letting x_i be the 1-dimensional position of agent i, the initial states are given by $x_1(0) = 0$, $x_2(0) = 5$, and $x_3(0) = 15$. Assume that all of the trust values initially start positive, at $\tau_i(0) = 1$.

As in [49], the states evolve according to a weighted gradient-descent scheme where the trust values are the weights, which, for the cost in (27), is given by

$$\dot{x}_i = \tau_i \sum_{j \in N_i} (x_j - x_i)$$

Because agent 2 is further from agent 3 than to agent 1, it will initially move to the right (and away from agent 1). Agent 1 will see this as agent 2 not contributing to minimizing their inter-agent cost F_{12}, and thus agent 1's trust toward agent 2 will decrease, according to

Figure 12. $N = 3$ **line graph with circles representing the initial positions of the agents.**

the self-centered trust model in (26). However, agent 2 is indeed contributing to minimizing the overall cost $F(x)$ by moving closer to agent 3. The limitation with the old trust model is that agent 1 does not take into account (or simply it is not aware of) what is going on in agent 2's neighborhood.

Since the goal is, however, to minimize the overall team cost $F(x)$, we need a new trust model that resolves this issue. Instead of using a self-centered trust model, we propose that each agent should look at its neighbors' *overall* contributions to decreases in the cost when updating their trust values, which will give us a "team-oriented" trust model. This requires that agents have 2-hop information, meaning they need information from their neighbors' neighbors to calculate trust. Because we claim that these dynamics represent those of humans, this requirement is reasonable in the sense that humans have less restrictive perception of the environment than robots. We will present this modified trust model in Section 4.2.1.

4.2.1 Team-Oriented Trust in Homogeneous Multi-Agent Networks

To begin with, we assume that all of the agents are "human-like" in that they all are capable of reasoning about trust and thus incorporate a trust metric into their dynamics. Let there be N agents, where $x_i \in \mathbb{R}^d$ is the state of agent i, which could be a position or an opinion for example, and $\tau_i \in \mathbb{R}$ is the trust that agent i has for its neighbors. If we stack the agents' states into a vector, we can represent the collective state as $x = [x_1^T, x_2^T, \ldots, x_N^T]^T \in \mathbb{R}^{Nd}$ and the collective trust as $\tau = [\tau_1, \ldots, \tau_N]^T \in \mathbb{R}^N$

In this work, we consider scenarios in which the desired network performance is specified by the following performance cost,

$$F(x) = \sum_{(i,j)\in E} F_{ij}(\|x_i - x_j\|) \tag{28}$$

where $F_{ij}(\|x_i - x_j\|)$ is the symmetric, pairwise performance cost, as is done widely in the multi-agent network literature, for example, for formation control [54–56]. The goal of the team of agents is to achieve a configuration that minimizes $F(x)$.

As motivated by the DeGroot model [57] and done in our previous work in [49] as well as in other works [34], we let the state dynamics of agent i evolve by weighting the standard gradient-descent approach by the trust that agent i has for its neighbors, that is

$$\dot{x}_i = -\tau_i \sum_{j\in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i}. \tag{29}$$

Note that we do not claim that human behavior is exactly modeled in this way, but that in a human-swarm system, the human would be willing to follow these dynamics. However, because of human cognition, the human may lose (or gain) faith in the system, causing the trust to decrease (or increase) and thus the human behavior to change accordingly.

However, the difference between the model in [49] and the one we introduce here is reflected in the trust dynamics, as discussed in Section 2.3. In this work we focus on "neighborhood" trust, meaning that agent i trusts all of its neighbors the same amount. Because we are concerned with how much each neighbor contributes to decreasing the overall cost, we note that the change in cost is given by

$$\frac{dF(x)}{dt} = \sum_{i=1}^{N} \frac{\partial F(x)}{\partial x_i}^T \dot{x}_i$$

and therefore the direct contribution of agent i to the *decrease* in cost, due to the movement of agent i, is $-\frac{\partial F(x)}{\partial x_i}^T \dot{x}_i$. Thus, the higher this term is, the more that agent i is contributing to the team goal, and thus the more that its neighbors should trust it.

To reflect this, we let the evolution of trust for agent i be the sum of its neighbor's contributions to the decrease in cost,

Team-Oriented Trust Model

$$\dot{\tau}_i = -\sum_{j \in N_i} \frac{1}{|N_j|} \frac{\partial F(x)}{\partial x_j}^T \dot{x}_j \tag{30}$$

or alternatively,

$$\dot{\tau}_i = -\sum_{j \in N_i} \left[\frac{1}{|N_j|} \left(\sum_{k \in N_j} \frac{\partial F_{kj}(\|x_k - x_j\|)}{\partial x_j} \right)^T \dot{x}_j \right] \tag{31}$$

where we added a scale factor of $1/|N_j|$ to the contribution by neighbor j, where $|N_j|$ is the neighborhood cardinality of agent j. This essentially "averages" out the contributions of agent j so that one agent does not have more of an influence on the change in trust just because it has more neighbors.

At this point, as done for the self-centered trust model, we could also define a pairwise trust value, τ_{ij} and similarly define the trust dynamics for it in a straightforward manner. We focus on the neighborhood trust model in this work and thus leave this out.

4.2.1.1 Invariance Results

If we define $\hat{\tau}$ to be the total trust in the system, that is

$$\hat{\tau} = 1^T \tau = \sum_{i=1}^{N} \tau_i, \tag{32}$$

then, similar to the self-centered trust modeling presented in [49], as a result of the coupled dynamics we get an invariance result relating the total trust to the performance of the system.

Theorem 7. *Consider an N-agent system with a static, connected interaction graph G, where the agents' state dynamics are given by* (29) *and the trust dynamics are given by* (31). *Consider a performance cost $F(x)$ and the total trust $\hat{\tau}$ as defined in* (28) *and* (32)*, respectively. Then, the following invariance holds*

$$\frac{d}{dt}\left(F(x) + \hat{\tau}\right) = 0.$$

Proof. To prove this result let us notice that

$$\dot{\hat{\tau}} = \sum_{i=1}^{N} \dot{\tau}_i = -\sum_{i=1}^{N} \sum_{j \in N_i} \frac{1}{|N_j|} \frac{\partial F(x)}{\partial x_j}^T \dot{x}_j$$

where the term

$$\frac{1}{|N_j|} \frac{\partial F(x)}{\partial x_j}^T \dot{x}_j$$

has no dependence on i and appears in the summation once for every time that agent j is a neighbor of an agent i, or exactly $|N_j|$ times. Consequently, the dynamics for $\hat{\tau}$ can be written as

$$\dot{\hat{\tau}} = -\sum_{i=1}^{N} \frac{\partial F(x)}{\partial x_i}^T \dot{x}_i. \tag{33}$$

Furthermore, if we write $F(x)$ from (28) in a slightly different manner, that is,

$$F(x) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in N_i} F_{ij}(\|x_i - x_j\|)$$

then the time derivative is given by

$$\frac{d}{dt} F(x) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j \in N_i} \left(\frac{\partial F_{ij}}{\partial x_i}^T \dot{x}_i + \frac{\partial F_{ij}}{\partial x_j}^T \dot{x}_j \right),$$

which, since the performance costs are symmetric, i.e. $F_{ij} = F_{ji}$, and the network is undirected, simplifies to

$$\frac{d}{dt} F(x) = \sum_{i=1}^{N} \sum_{j \in N_i} \frac{\partial F_{ij}}{\partial x_i}^T \dot{x}_i. \tag{34}$$

and by substituting

$$\frac{\partial F(x)}{\partial x_i} = \sum_{j \in N_i} \frac{\partial F_{ij}}{\partial x_i}$$

into the expression for $\dot{\hat{\tau}}$ in (33), we get $\frac{d}{dt} F(x) = -\dot{\hat{\tau}}$ and thus the desired result is obtained.

■

What this result tells us is that in order to achieve the desired performance, i.e. minimizing the cost $F(x)$, we must have that the total trust in the system increases. In fact, the two are intimately linked, meaning that decreases in total trust correspond to increases in the cost, or decreased performance. This is consistent with the organizational psychology literature in which studies have shown a correlation between trust and performance in teams within organizations [62].

4.2.1.2 Convergence Properties

One of the results that was shown in Section 4.1 was that, for two agents executing rendezvous, we know under what specific conditions on the initial trust that the agents will achieve their goal and under what conditions the states will diverge. However, for the self-centered trust model, it was not possible to show more general convergence guarantees for more than two agents. In fact, the two-agent scenario is a special case in that it results in the self-centered trust model being exactly the same as the team-oriented trust model because there is only one edge in the graph. With the new, team-oriented trust model, however, we can give explicit conditions under which the system will achieve the goal of minimizing the performance cost. In order to show the convergence results, we first discuss some basic monotonicity properties of the agents' trust values.

Lemma 6. *Consider an N-agent system where the agents' state dynamics are given in* (29) *and the trust dynamics are given in* (31)*. Assume that the static interaction graph G is connected and that* $\tau_i(0) > 0$*, for all* $i \in \{1, \dots, N\}$*. Then* τ_i *is non-decreasing and* $\tau_i(t) \geq \tau_i(0)$*, for all* $i \in \{1, \dots, N\}$ *and for all* $t > 0$*.*

Proof. We can rewrite (29) for agent j as

$$\dot{x}_j = -\tau_j \frac{\partial F(x)}{\partial x_j}$$

and substitute this into (30) to get the following expression for the trust dynamics,

$$\dot{\tau}_i = \sum_{j \in N_i} \frac{\tau_j}{|N_j|} \frac{\partial F(x)}{\partial x_j}^T \frac{\partial F(x)}{\partial x_j}$$

or, equivalently,

$$\dot{\tau}_i = \sum_{j \in N_i} \frac{\tau_j}{|N_j|} \left\| \frac{\partial F(x)}{\partial x_j} \right\|^2 .$$

Since $\tau_j(0) > 0$, for all $j \in \{1, \dots, N\}$, $\dot{\tau}_i \geq 0$, for all $i \in \{1, \dots, N\}$, and it follows that $\tau_i(t) \geq \tau_i(0) > 0$, for all $i \in \{1, \dots, N\}$, for all $t > 0$. ∎

This tells us that if the initial trust values are all positive, they will stay positive over the duration of the evolution of the dynamics and also finite due to the invariance in Theorem 7. Similarly, if all of the trust values are initially negative, they will remain negative, as shown in the following lemma.

Lemma 7. *Consider an N-agent system where the agents' state dynamics and trust dynamics are given in* (29) *and* (31)*, respectively. Assume that the static interaction graph G is connected and* $\tau_i(0) < 0$*, for all* $i \in \{1, \dots, N\}$*. Then* τ_i *is non-increasing and* $\tau_i(t) \leq \tau_i(0)$*, for all* $i \in \{1, \dots, N\}$ *and for all* $t > 0$*.*

Proof. This proof follows directly from the proof of Lemma 6 by simply flipping the inequalities. ■

We are now ready to discuss convergence results of the objective for the multi-agent system. First we introduce $\Gamma = \{x^* \mid dF(x^*)/dx = 0\}$, the set of all critical points of F.

Theorem 8. *Consider an N-agent system with the agents' state dynamics in* (29) *and the trust dynamics in* (31)*. Assume the underlying static interaction graph G is connected and* $\tau_i(0) > 0$*, for all* $i \in \{1, \dots, N\}$*. Consider a performance cost* $F(x)$ *and the total trust* τ *as defined in* (28) *and* (32)*, respectively. Then* $x(t)$ *asymptotically converges to* Γ*.*

Proof. The collective agent dynamics are given by

$$\dot{x}(t) = -B(\tau(t))\frac{dF(x(t))}{dx} \tag{35}$$

where $B(\tau(t)) = \text{diag}(\tau(t)) \otimes I_d$ and $\text{diag}(\tau(t)) \in \mathbb{R}^{N \times N}$ is a diagonal matrix with $\tau_i(t)$ as its i'th diagonal element, I_d is the $d \times d$ identity matrix, and $\otimes$ is the Kronecker product.

Taking the time derivative of the performance cost yields

$$\frac{dF(x)}{dt} = \frac{dF(x)}{dx}^T \dot{x} = -\frac{dF(x)}{dx}^T B(\tau(t))\frac{dF(x)}{dx} \tag{36}$$

and, following the result from Lemma 6, $\tau_i(t) \geq \tau_i(0) > 0$, for all $i \in 1, \dots, N$. Therefore, $B(\tau(t))$ has all positive elements on the diagonal and thus is positive definite for all t, giving

the desired result $\frac{dF}{dt} < 0$ for all $\left\|\frac{dF(x(t))}{dx}\right\| \neq 0$, and $\frac{dF}{dt} = 0$ if and only if $\left\|\frac{dF(x(t))}{dx}\right\| = 0$. At this point, since the agents are moving along the anti-gradient of the cost function and $\dot{x} = 0$ only when $\frac{dF(x)}{dx} = 0$, it follows that $x(t)$ converges to Γ. ■

Here we note that the only way the system will end up at a local maximum is if the initial states are such that it starts on one. Therefore, it will mostly reach local minima or saddle points. However, because minima are locally stable equilibrium points while saddle points are not, the system will typically reach a local minimum.

This result is straightforward in that the model was designed so that trust will increase as long as all agents are behaving correctly and as long as the trust values stay positive, the state dynamics are designed such that the agents will indeed behave correctly. We note that this result follows from the ideal, nominal case in which every agent has perfect information about its neighbors and that all agents behave according to the state dynamics in (29). Things become more interesting when agents do not have perfect information about its neighbors' states and when there are noisy measurements and corruption and possibly deceit added to the system. Future work will be devoted to exploring these avenues.

The same type of analysis can be done for a system where all of the trust values are initially negative, showing that the agents will not reach a configuration that corresponds to a local minimum of the cost function $F(x)$ and may actually cause the agents' state trajectories to diverge.

Theorem 9. *Consider an N-agent system where the agents' state dynamics are given in* (29) *and the trust dynamics are given in* (31)*. Assume the static interaction graph G is connected and* $\tau_i(0) < 0$*, for all* $i \in 1, \ldots, N$*. Consider a performance cost* $F(x)$ *and the total trust* $\hat{\tau}$ *as defined in* (28) *and* (32)*, respectively. Then the agents will not reach a configuration corresponding to a local minimum of the performance cost* $F(x)$*.*

Proof. Following Lemma 7, we know that $\tau_i(t) < 0$ for all $t \geq 0$ and all $i \in \{1, \ldots, N\}$. Hence, $B(\tau(t))$ in (35) is negative definite for all $t \geq 0$ and from (36), we know that $\frac{dF}{dt} > 0$

for all $\left\|\frac{dF(x)}{dx}\right\| \neq 0$. And, because every agent is updating its state in the direction corresponding to an increase in $F(x)$, two cases may arise according to the particular nature of the cost function $F(x)$ and the initial configuration of the system, that is either the agents will stop in a configuration corresponding to a local maximum for which $\left\|\frac{dF(x)}{dx}\right\| = 0$, or the agents' states will keep updating indefinitely while the performance cost goes to infinity. Clearly, in either of these cases the agents will not reach a local minimum of $F(x)$. ■

These theorems allow us to predict the behavior of the system when either all of the trust values are initially positive or all of them are initially negative. This suggests that trust systems should be initiated with positive trust in order to guarantee that the desired performance is achieved. Note that, if the initial trust values of the agents are mixed, i.e., both positive and negative, then following the previous analysis, we cannot make any claim about the definiteness of the matrix $B(\tau(t))$, and thus about the convergence properties of the system.

4.2.2 Team-Oriented Trust in Heterogeneous Multi-Agent Networks

As part of our motivation for this work was to analyze trust in human-swarm interaction scenarios, let us now consider a more general scenario where only a portion of the N agents, i.e., human beings, are driven by a trust attitude while the remaining agents, i.e., robots, act "classically" according to a standard gradient-descent approach. N_H is the number of so-called "human" agents that operate according to a trust metric, and the first N_H states represent the states of these agents. Let us define the two sets of indices $H = \{1, 2, \ldots, N_H\}$ and $R = \{N_{H+1}, \ldots, N\}$. Then, for $i \in H$, the dynamics evolve according to (29), repeated here for ease of reference

$$\dot{x}_i = -\tau_i \sum_{j \in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i} \tag{37}$$

where the evolution of the trust values, τ_i for $i \in H$, are given in (30).

The rest of the agents are the so-called "robots" that are autonomous and do not operate according to trust metrics. The states of these agents are given by x_i for $i \in R$, with

dynamics given by

$$\dot{x}_i = -\sum_{j \in N_i} \frac{\partial F_{ij}(\|x_i - x_j\|)}{\partial x_i} \tag{38}$$

which is the standard gradient-descent based method without the trust multipliers, as only the human agents are trusting entities.

We can now demonstrate that the convergence results shown in Section 4.2.1.2 for the homogenous dynamics still holds for the heterogeneous dynamics under the same working conditions, i.e., as long as the initial trust values are positive, the system will converge to a configuration that minimizes the performance cost $F(x)$.

To this end, let us first discuss some similar basic monotonicity properties of the agents trust values as in Lemma 6 for this heterogeneous setting.

Lemma 8. *Consider a collection of N agents under the state dynamics in* (37) *for $i \in H$ and by* (38) *for $i \in R$ and the trust dynamics in* (30) *for $i \in H$. Assume that the underlying static interaction graph G is connected and $\tau_i(0) > 0$, for all $i \in H$. Then $\tau_i(t) \geq \tau_i(0)$, for all $i \in H$ and for all $t > 0$.*

Proof. This result follows directly from the proof of Lemma 6 where the only difference lies in the fact that some of the agents no longer have trust dynamics. In particular, by recalling that for the human agents, i.e. $i \in H$, trust evolves according to

$$\dot{\tau}_i = -\sum_{j \in N_i} \frac{1}{|N_j|} \frac{\partial F(x)}{\partial x_j}^T \dot{x}_j$$

where we can split up N_i into agent i's neighbors that are robots and the ones that are humans and substitute in $\dot{x}_j$ for the corresponding type of agent, i.e.,

$$\dot{\tau}_i = \sum_{j \in N_i \cap H} \frac{\tau_j}{|N_j|} \left\| \frac{\partial F(x)}{\partial x_j} \right\|^2 + \sum_{j \in N_i \cap R} \frac{1}{|N_j|} \left\| \frac{\partial F(x)}{\partial x_j} \right\|^2 \tag{39}$$

Since $\tau_j(0) > 0$ for all $j \in H$, we get that $\dot{\tau}_i \geq 0$ for all $i \in H$, and thus $\tau_i(t) \geq \tau_i(0) > 0$ for all t and all $i \in H$. ∎

To prove convergence of this system, we additionally need to show that the trust values remain finite, i.e. they do not go to infinity. In order to do this, we give a similar result to the invariance result from the homogeneous case. If we again define $\hat{\tau}$ to be the total trust in the system, that is

$$\hat{\tau} = \sum_{i \in H} \tau_i,$$

then we observe the following result.

Lemma 9. *Consider a collection of N agents under the state dynamics in* (37) *for $i \in H$ and by* (38) *for $i \in R$ and the trust dynamics in* (30) *for $i \in H$. Assume that the underlying static interaction graph G is connected and $\tau_i(0) > 0$, for all $i \in H$. Then*

$$\frac{d}{dt}\left(F(x) + \hat{\tau}\right) \leq 0.$$

Proof. To prove this result let us notice that

$$\dot{\hat{\tau}} = \sum_{i \in H} \dot{\tau}_i$$

which, after plugging in $\dot{\tau}_i$ from (39), can be rewritten as

$$\dot{\hat{\tau}} = \sum_{i \in H} \frac{d_i \tau_i}{|N_i|} \left\| \frac{\partial F(x)}{\partial x_i} \right\|^2 + \sum_{i \in R} \frac{d_i}{|N_i|} \left\| \frac{\partial F(x)}{\partial x_i} \right\|^2$$

where $d_i = |N_i \cap H|$ is the number of human agents that agent i has in its neighborhood set.

Furthermore, the time derivative of $F(x)$ is given by

$$\frac{d}{dt} F(x) = -\sum_{i \in H} \tau_i \left\| \frac{\partial F(x)}{\partial x_i} \right\|^2 - \sum_{i \in R} \left\| \frac{\partial F(x)}{\partial x_i} \right\|^2$$

and by noting that $d_i \leq |N_i|$ for all i and $\tau_i(t) > 0$ for all $i \in H$, we get $\frac{d}{dt} F(x) + \dot{\hat{\tau}} \leq 0$. ∎

We can now prove a convergence result for this heterogeneous scenario similar to the one given in Theorem 8 for the homogeneous setting.

Theorem 10. *Consider a collection of N agents under the state dynamics in* (37) *for $i \in H$ and by* (38) *for $i \in R$ and the trust dynamics in* (30) *for $i \in H$. Assume that the underlying static interaction graph G is connected and $\tau_i(0) > 0$ for all $i \in H$. Then the states $x(t)$ asymptotically converge to the set of critical points,* Γ.

Proof. This result follows the proof of Theorem 8 where the only difference lies in the fact that some of the agents no longer have trust dynamics. If we evaluate the collective dynamics of the system, we get

$$\dot{x}(t) = -\begin{bmatrix} C(\tau(t)) & 0 \\ 0 & I_{N_R d} \end{bmatrix} \frac{dF(x(t))}{dx} \tag{40}$$

where $C(\tau(t)) = \text{diag}(\tau_1(t), \ldots, \tau_{N_H}(t)) \otimes I_d$ where I_d is the $d \times d$ identity matrix, and $\otimes$ is the Kronecker product.

Taking the time derivative of the performance cost $F(x)$ yields

$$\frac{dF(x(t))}{dt} = -\frac{dF(x(t))}{dx}^T \begin{bmatrix} C(\tau(t)) & 0 \\ 0 & I_{N_R d} \end{bmatrix} \frac{dF(x(t))}{dx} \tag{41}$$

where, because $\tau_i(t) > 0$ for all t and $i \in H$, then $C(\tau(t))$ is positive definite for all t, implying that $\text{diag}(C(\tau(t)), I_{N_R d})$ is also positive definite. Thus, $\frac{dF}{dt} < 0$ for all $\left\| \frac{dF(x(t))}{dx} \right\| \neq 0$, and $\frac{dF}{dt} = 0$ if and only if $\left\| \frac{dF(x(t))}{dx} \right\| = 0$. Thus, $x(t)$ converges to Γ. ∎

Consequently, having heterogeneous dynamics where some agents no longer operate according to a trust metric does not break the convergence guarantees that we had for the homogeneous network dynamics. In fact, it could be intuitively seen as replacing humans with robots where the robots are always going to do the "right" thing and thus as long as the human agents start with positive initial trust, the entire system still works as expected.

4.2.3 Simulations

In this section, we present simulation results to corroborate the theoretical findings. As a case study of the gradient-descent multi-agent framework discussed in previously, we consider the seminal swarm aggregation work described in [64]. Swarm aggregation has been widely investigated by the robotics and control communities over the last two decades, e.g., [65–67] and is a basic behavior that many swarms in nature exhibit, such as some bacteria, ant colonies, bee colonies, flocks of birds, and schools of fish. Moreover, many of the collective behaviors seen in biological swarms and some behaviors to be possibly implemented in engineering multi-agent dynamic systems emerge in aggregated swarms. The

reader is referred to [68] for a comprehensive overview of swarm stability and optimization. In particular, in our setting we are assuming the swarm may be either homogeneous, i.e., all agents have trust dynamics, or heterogeneous, i.e., only a subset of the agents have trust dynamics.

As in [68], we consider the following performance cost $F(x)$ in order to achieve aggregation

$$\begin{aligned} F(x) &= \sum_{i=1}^{N} \sum_{j \in N_i} F_{ij}(x) \\ &= \sum_{i=1}^{N} \sum_{j \in N_i} [F_{\mathrm{a}}(\|x_i - x_j\|) - F_{\mathrm{r}}(\|x_i - x_j\|)] \end{aligned} \tag{42}$$

where $F_{\mathrm{a}}(\|x_i - x_j\|)$ and $F_{\mathrm{r}}(\|x_i - x_j\|)$ are the pairwise aggregate performance cost and repulsive performance cost, respectively. In particular, the following pairwise aggregate and repulsive performance costs have been used

$$F_{\mathrm{a}}(\|x_i - x_j\|) = a \, \frac{\|x_i - x_j\|^2}{2} \tag{43}$$

and

$$F_{\mathrm{r}}(\|x_i - x_j\|) = b \, \log(\|x_i - x_j\|) \tag{44}$$

for which we have the following pairwise gradients

$$\frac{\partial F_{\mathrm{a}}(\|x_i - x_j\|)}{\partial x_i} = a \, (x_i - x_j) \tag{45}$$

and

$$\frac{\partial F_{\mathrm{r}}(\|x_i - x_j\|)}{\partial x_i} = b \, \frac{(x_i - x_j)}{\|x_i - x_j\|^2} \tag{46}$$

with a and b the aggregation and repulsion tuning parameters, respectively, which determines the size of the aggregation area (see again [68] for further details).

For purposes of the simulations in this section, we let the constants $a = 1$ and $b = 2$ and used 2-dimensional agents, i.e. $d = 2$. We first simulated the dynamics for the homogeneous case with $N = 6$ where all initial trust values are positive. The interaction graph

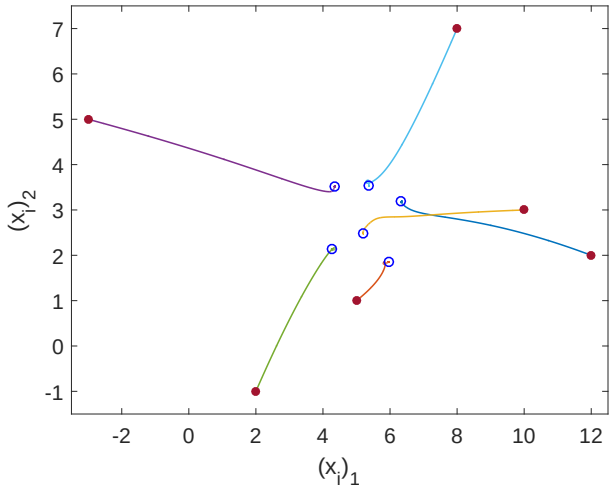


Figure 13. State trajectories for the homogeneous trust-based algorithm with all positive initial trust values. The filled circles represent the initial states and the empty circles represent the states at the end of the simulation.

$G = (V, E)$ is defined such that $E = E_K \setminus \{(v_1, v_5), (v_2, v_4)\}$ where E_K is the edge set associated with a complete graph, G_K. The initial stacked states are $\tau(0) = [1, 2, 0.5, 2, 6, 0.2]^T$ and $x(0) = [12, 2, 5, 1, 10, 3, -3, 5, 2, -1, 8, 7]^T$. Because all of the trust values are initially positive, this system should converge to a minimum of $F(x)$, due to Theorem 8, and it does, as shown in Figure 4.2.3.

If we keep the initial states and the interaction graph the same but instead change this to a heterogeneous case where we let only agents 3 and 5 be human agents with trust metrics and let the rest be robots, we get the trajectories seen in Figure 14. Here, $\tau_3(0) = 2$ and $\tau_5(0) = 0.5$ (both positive) and therefore the result of Theorem 10 holds and the states converge to a configuration pertaining to a minima of the cost. Notably, the trajectories for this scenario are different than those in Figure 4.2.3, as the dynamics are clearly different, but both reach a local minimum of the cost.

We also simulated the homogeneous dynamics with negative initial trust values to illustrate the negative result of Theorem 9. Here, $N = 3$, the graph $G = G_K$ is complete, and the initial states are $x(0) = [2, -3, 6, 10, 12, 8]^T$ and $\tau(0) = [-2, -0.5, -1]^T$. Figure 15 shows the evolution of the states for the first 65 time steps in the simulation, where it can be seen

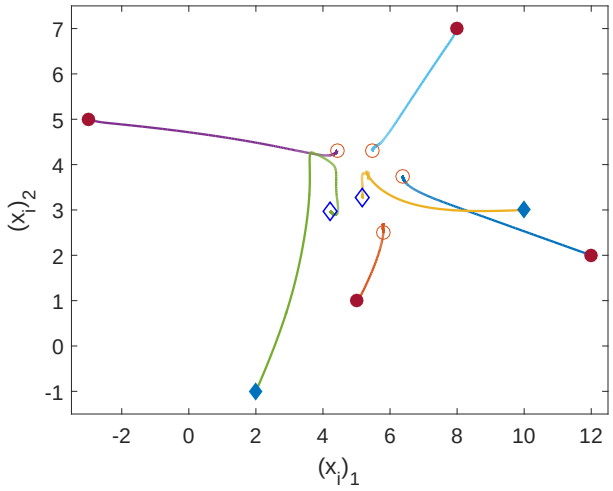


Figure 14. State trajectories for the heterogeneous trust-based algorithm, where agents 3 and 5 have positive initial trust and the rest are robots without trust metrics. The filled circles (diamonds) represent the initial states of the robot (human) agents and the empty circles (diamonds) represent the robot (human) agents' states at the end of the simulation.

that the agents are moving away from each other, and they continue to do that for the times not pictured. In fact, the cost $F(x)$ blows up over time and thus a minima is not reached.

4.3 Trust-Action Couplings

In the past two sections we presented models in which agents interact and exchange information in order to make collective decisions. In the models presented, external influences are not considered – that is, the evolution of an agent's state is dictated only by that agent's state and its neighbors' states. However, sometimes external influences, e.g. due to the environment, may act upon the agents, introducing a drift term in the consensus-based algorithms. This type of model is known as the coupled drift-diffusion model (DDM) and has been used in the literature for modeling collective decision-making in ideal human groups as well as animal groups, e.g. [69]. The drift-diffusion model is known for being well-equipped to model human behavior in two-alternative choice tasks [70], in which evidence is agreggated and people choose one of two alternatives dictated by the evidence crossing a threshold [71].

Sometimes, the decision that a person or animal makes may lead to an action. For

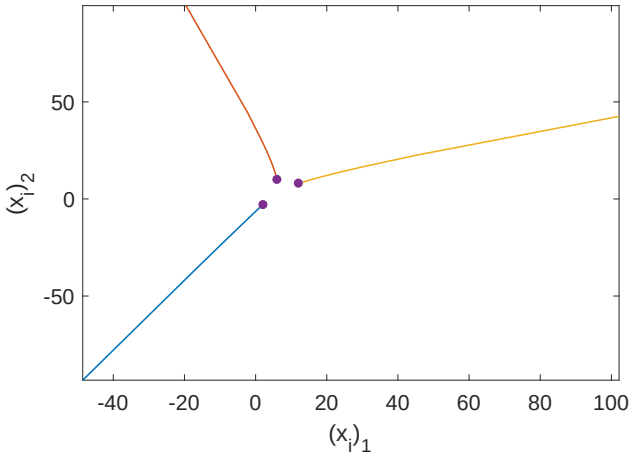


Figure 15. State trajectories for the homogeneous trust-based algorithm with all agents having negative initial trust. The filled circles represent the initial states and the trajectories are given for $t = 0$ to $t = 65$ (simulation time).

example, if there is a group of birds accumulating evidence in order to decide whether to leave a stopover site, the decision being made could result in an action, i.e. a bird taking off in flight. We are interested in determining how this action affects the decision making processes of the other birds that have not yet made a decision, which has not been studied in the literature. If a bird decides to take-off, this action may provoke the others to leave as well. Alternatively, it may come back down if no other birds follow.

We can also think of this in terms of human group decision-making. For example, if a group of people comes up to an intersection and is deciding whether to walk left or to walk right, they may converse with each other first to try to decide upon a direction before they actually start walking. However, if one person makes a decision before the others, and starts walking left, this action may propagate and influence the decisions of the others. For example, the others could decide to follow the one who started walking. Similarly, the person who made the decision to start walking may look back and see if anyone else is following, and could modify his or her decision based upon the actions of the others in the group. Furthermore, these outcomes could vary depending on how much trust the others have in the person that started walking away.

To capture this type of behavior, we explore a particular manifestation of this, whereby the couplings between decision making, actions, and trust are taken into consideration. We begin by introducing a model that connects decision making and actions, using migrating birds as inspiration, and then later incorporate trust.

4.3.1 Decision Making and Actions

As an illustration, we first introduce a model that captures the aforementioned interplay between decision making and actions, using migrating birds as the basis for this work. It has been observed that some species of birds display pre-flight *intention movements*, such as wing flapping, as a signal to other birds in the vicinity that it is ready to take-off in flight [72]. And, if others do not join in on this behavior, the birds that initiated usually will not fly [73], most likely due to the benefits inherent in migrating in flocks, such as better protection against predators [74]. In this work, we develop a model that captures these social interactions between birds that are deciding whether to take off in flight in order to better understand the connections between decision making and actions.

4.3.1.1 Dynamical Model

Consider a network of N agents, each with a state, $x_i \in \mathbb{R}$. Because the behavior of migrating birds is the motivation for this work, we can think of this state as the intensity of a bird's pre-flight intention movements, or how much it would like to take off in flight. Each agent can be in one of three states, namely Ground, Air, or Gone. All of the agents are initialized to the Ground state and once an agent's accumulated evidence reaches a certain threshold, ρ, an action is taken, causing the agent to transition to the Air state. This is the equivalent of a bird's intention movements being intense enough to cause it to take off in flight. And, once an agent has been in the Air state for longer than ΔT seconds, it transitions to the Gone state. However, an agent can come back to the Ground state, if its accumulated evidence becomes less than ρ minus some small offset ϵ. This is representative of a bird coming back down to the ground if the other birds do not follow it. The state diagram for a single agent i is shown in Figure 16.

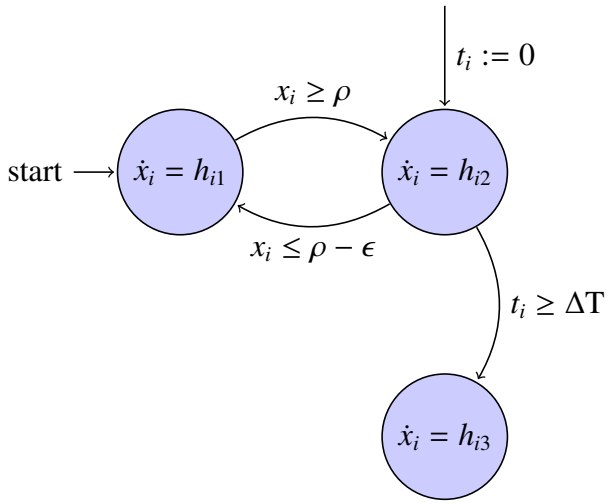


Figure 16. State diagram for agent i

The dynamics of the agent depend on which state it is in and evolve according to h_{i1}, h_{i2}, or h_{i3}. When two neighboring agents are both in the Ground state or both in the Air state, they share their information with each other and the weight on this information diffusion is 1. Coming back to the bird scenario, this represents birds being able to see (or hear) the social cues from the other birds in the form of the intention movements or calls. When, instead, one agent is in the Ground state and its neighbor is in the Air state, the weight, γ, on the information diffusion is greater than 1 to enforce cohesion in a stronger manner. Each agent (or bird) can sense environmental cues causing it to be more or less inclined to leave, and this is represented by a drift term, f_i.

If agent i is in the Ground state, then $i \in G$, and if agent i is in the Air state, then $i \in A$, where $G, A \subseteq \{1, 2, \ldots, N\}$. We introduce an action state $a_i(t) \in \{0, 1, 2\}$ such that $a_i = 0$ if $i \in G$, $a_i = 1$ if $i \in A$, and $a_i = 2$ if agent i is in the Gone state. In the Ground state, the dynamics of agent i evolve according to h_{i1}, given by

$$\dot{x}_i = h_{i1} = f_i + \sum_{j \in N_i \cap G} (x_j - x_i) + \gamma \sum_{j \in N_i \cap A} (x_j - x_i). \tag{47}$$

The dynamics change when agent i enters the Air state, where now agent i's state evolves according to

$$\dot{x}_i = h_{i2} = f_i + \gamma \sum_{j \in N_i \cap G} (x_j - x_i) + \sum_{j \in N_i \cap A} (x_j - x_i). \tag{48}$$

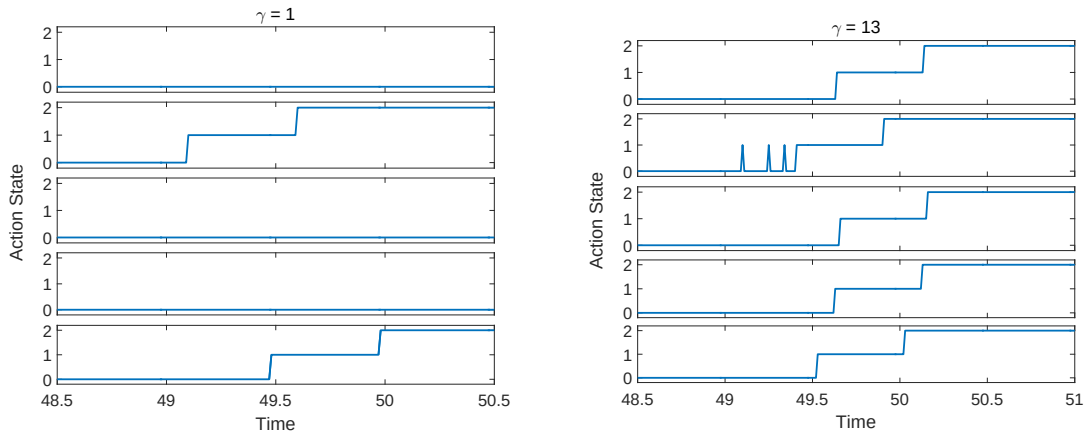


Figure 17. Action profiles for 5 agents executing the coupled decision making-actions algorithm. Ground is represented by 0, Air by 1, and Gone by 2. In the left figure, $\gamma = 1$ and in the right figure, $\gamma = 13$.

Here N_i is the neighborhood set of agent i which represents the agents with whom agent i can share information, or in the bird scenario, which agents can sense agent i's social cues. As mentioned previously, once an agent is in the Air state for more than ΔT seconds, it leaves the group and is considered in the Gone state. In this state, the agent is considered no longer in the network and its state no longer evolves, that is, $\dot{x}_i = h_{i3} = 0$.

4.3.1.2 Simulations

With the model fully developed, we present simulations in order to show how the actions taken affect the decision making process. We let the threshold $\rho = 50$, $\Delta\text{T} = 0.5$, and $\epsilon = 0.1$. If we stack the states into a vector and define $x(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$ and similarly stack the environmental drivers into a vector $f = [f_1, f_2, \ldots, f_N]^T$, then the initial conditions used for the simulations are $x(0) = [23.46, 6.39, 13.33, 23.87, 6.69]^T$ and $f = [0.0003, 2.7107, -0.7257, 0.1976, 1.3663]^T$. The information-exchange graph was chosen to be complete, that is, every agent shares information with every other agent. In other words, $N_i = \{1, 2, \ldots, N\} \setminus \{i\}$, for all i. The dynamics in (47) and (48) were simulated using these initial conditions for both $\gamma = 1$ and $\gamma = 13$.

When $\gamma = 1$, an agent weighs all of its neighbors' states the same, regardless of whether they are in the Ground state or in the Air state. As can be seen on the left-hand side of

Figure 17, agents 1, 3, and 4 remain in the Ground state, while agents 2 and 5 end up in the Gone state. Therefore, the group does not remain as a cohesive unit. However, when we let $\gamma = 13$, all agents end up in the Gone state within 0.25 seconds of each other, as seen in the right-hand plot of Figure 17. As illustrated in the right-hand plot of Figure 17 and in the first four simulation snapshots in Figure 18, agent 2 switches between the Ground and Air states several times before eventually staying in the Air state, and later transitioning to the Gone state. This switching is consistent with bird migration, where some species exhibit repeated take-offs and landings before finally taking off as a group [72].

In order to analyze how quickly decisions are made, we define $T_{Gone} \in \mathbb{R}$ to be the amount of time it takes for all of the agents to enter the Gone state, that is

$$T_{Gone} = \inf \{t \in \mathbb{R}_{\geq 0} \,|\, a_i(t) = 2 \ \forall i \in \{1, 2, \ldots N\}\}.$$

As shown in Figure 18, when $\gamma = 13$, all of the agents are in the Gone state at t = 50.16 s, and because this is the first instance in time that this occurs, $T_{Gone} = 50.16$ s. When $\gamma = 1$, the simulation did not result in all agents entering the Gone state and thus T_{Gone} cannot be computed.

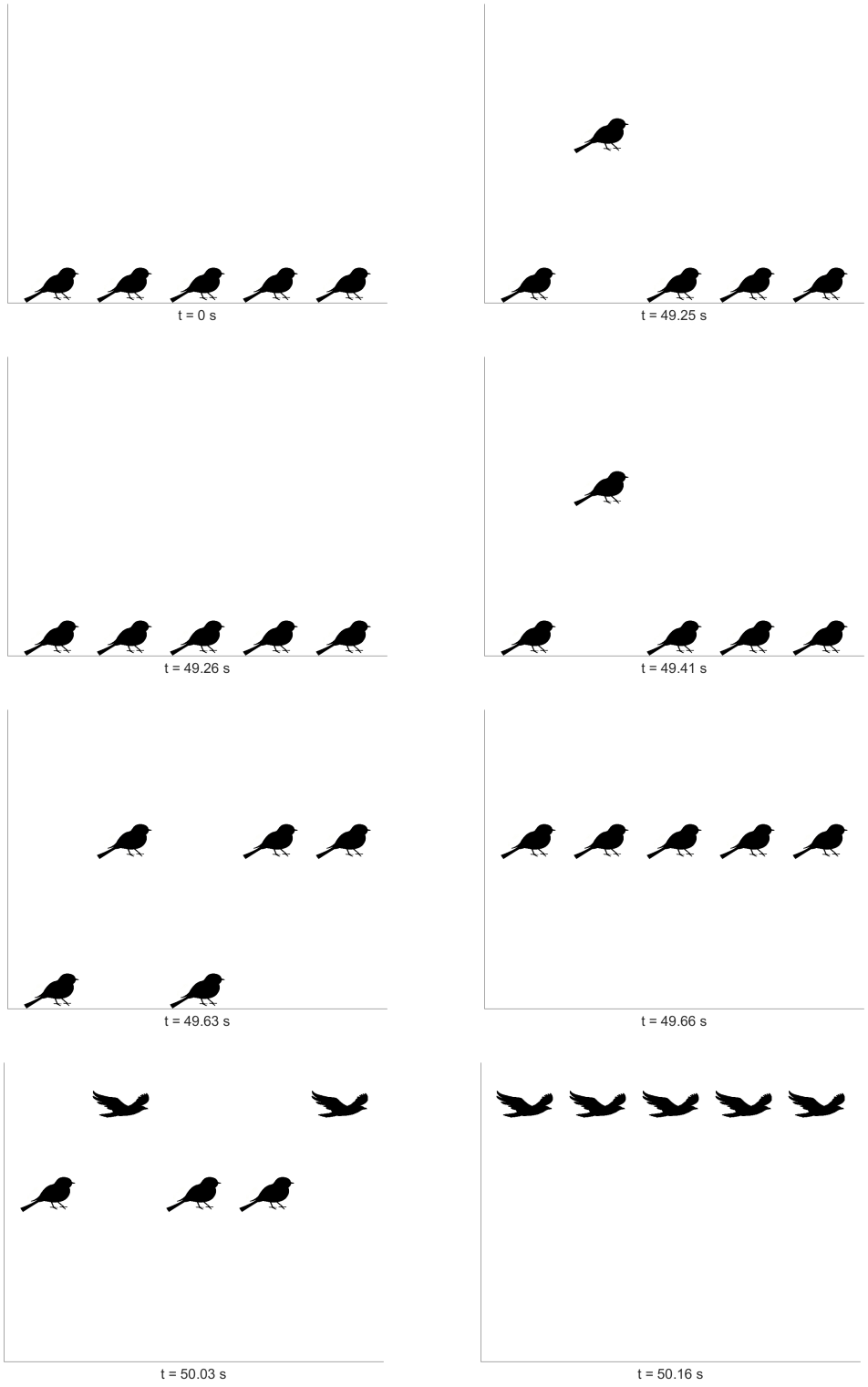


Figure 18. Simulation of 5 agents executing the coupled decision making-actions algorithm, with $\gamma = 13$. All of the birds start in the Ground state at t = 0 s, they all reach the Air state by t = 49.66 s, and are all in the Gone state at t = 50.16 s.

4.3.2 Towards Incorporating Trust

As mentioned previously, trust could also play a factor in the decision making process when actions are involved. Returning to the example of people deciding which direction to walk, if one person takes an action (i.e., walks away), the others may be more likely to follow him if they trust him. If they do not trust him, the action may have little influence or it may have the opposite effect and make the rest of the group more inclined to take an alternate action. One way in which trust could be incorporated into the model given in (47) and (48) is by replacing γ with a static, pair-wise trust value, τ_{ij}, representing the amount of trust that agent i has for agent j. The new dynamics for agent i become

$$\dot{x}_i = h_{i1} = f_i + \sum_{j \in N_i \cap G} (x_j - x_i) + \sum_{j \in N_i \cap A} \tau_{ij}(x_j - x_i) \tag{49}$$

when agent i is in the Ground state, and

$$\dot{x}_i = h_{i2} = f_i + \sum_{j \in N_i \cap G} \tau_{ij}(x_j - x_i) + \sum_{j \in N_i \cap A} (x_j - x_i) \tag{50}$$

when agent i is in the Air state.

In order to capture the fact that higher trust should lead to a person being more willing to follow the person that took an action, τ_{ij} should be positive and greater than 1 when trust exists and increase with increasing levels of trust. In the absence of trust, τ_{ij} could either be exactly 1 so that the action by agent j does not change the behavior of agent i, or it could be negative to reflect that distrust may cause agent i to be inclined to take a different action than that of agent j. It should also be noted that τ_{ij} does not have to be the same as τ_{ji}, that is, agent i may trust agent j more (or less) than agent j trusts agent i.

In [75], it is said that "...if there is trust among the members of the group then sometimes action may be taken without going through the decision making process so emphasizing the importance of trust." Although the model we introduced does not necessarily skip the decision making process, the existence of trust could speed up the process, as has been pointed out in the psychology literature, e.g. [76], which states that trust within organizations can improve the speed of decision making.

4.4 Conclusions

In this section, we introduced trust and state dynamics for multi-agent coordination that gives rise to a system that exhibits three properties observed in psychology: group polarization, belief polarization, and a positive trust-performance correlation. The trust notion, as well as the trust-state coupling, were developed to model the dynamics of a system where humans and autonomous agents coexist and interact. We showed that the total trust in the network is intimately linked to the performance of the system through an invariance and demonstrated that such a combination of social dynamics and physical update laws has the potential to make the performance of the system deteriorate dramatically.

Furthermore, we compared the rate of convergence of our model to that of the standard consensus protocol and also gave a closed-form expression for where the agents will meet under our dynamical model and compared it to the meeting point under standard consensus. We also introduced heterogeneous dynamics to capture differing behaviors of human and robot agents and analyzed the convergence of this system. Finally, we developed a model to capture the connections between trust, decision making, and actions in multi-agent systems and showed our results to be consistent with the organizational psychology literature.

CHAPTER 5

ENERGY-AWARE MULTI-ROBOT COORDINATION

As shown in the last chapter, trust has a large impact on *where* a group of humans and robots meet, or alternatively, what opinion they eventually agree upon. Moreover, depending on the initial trust levels, this meeting point may be significantly different than the average of the initial states or opinions. In fact, it may even be outside of the convex hull of the initial states of the agents. In this chapter, we explore an application area that results in a similar change in the meeting point, this time due to robots having different initial energy, or battery levels. Instead of robots meeting at the average of their initial positions, they meet at a point that depends on the initial battery levels and positions of the agents. The work in this chapter was published in [77] and [78].

5.1 Background and Problem Formulation

One driving application for multi-agent robotics is sustained environmental monitoring, where deployments are envisioned to take place in environments where it may not be easy to replace batteries or refill the fuel, such as hostile or inaccessible environments, e.g., [2, 79]. As a result, energy consumption must be taken into account during the design phase to ensure that the robot team can complete its mission.

In fact, numerous energy-aware algorithms have been developed in the sensor networks community, where the goal is to maximize the lifetime of the network while still satisfying some target level of coverage, e.g., [80–85]. These strategies come in to play once the network has been assembled. However, there is a broad class of problems in which mobility becomes the bottleneck. As a general observation, mobility is the most "expensive" when it comes to energy consumption, whereas communications are less costly, and computations and sensing are the least expensive, in comparison [86].

Previous efforts have been made to minimize energy consumption due to both mobility

and communication in multi-agent applications, e.g., [87] minimizes motion and communication energy in the context of transmitting information from a stationary object to a stationary remote station by having each agent act as a relay. In some cases, sensing energy is considered as well, e.g., in [88], mobility, sensing and communication costs were all taken into account from an energy-consumption point-of-view, in the context of a dynamic coverage problem.

We are interested in making energy consumption a central aspect of the coordination problem and instantiate this idea in the context of the rendezvous problem [52], in which a group of mobile robots are to meet at a common location using only relative displacement information. Previous work that couples energy to the rendezvous problem was presented in [86, 89], where the effects of shrinking sensing footprints, due to mobility-driven power decay, are analyzed in the context of a particular control law. Our work builds on this idea of connecting mobility-driven energy consumption to the rendezvous problem, but does so during the design phase, as opposed to analyzing the effects of energy consumption on a particular control law. We treat battery levels as a hard constraint and develop an algorithm for allowing the robots to meet in the least amount of time while ensuring that they do not run out of battery life.

As an illustration, suppose there are two robots that have the same initial battery level. In order to meet in an equitable way, a reasonable approach would be to have the robots meet at the average of their initial positions, as shown in Figure 19(a). Instead, suppose that they have different initial battery levels – now where should they meet? It seems intuitive that they should meet closer to the robot with a lower initial battery level, as shown in Figure 19(b). One primary objective of this research was to take this informal observation, apply it to a multi-robot scenario, and make it rigorous.

In order to solve the so-called energy-constrained spatio-temporal rendezvous problem, we separate it into two sub-problems, with the first being to determine where and when the mobile robots should meet, while satisfying the requirement that each robot must end up

(a) (b)

Figure 19. Two robots with the same initial battery levels should meet halfway (a), while, if the robot to the right has a lower initial battery level than the robot to the left, they should meet closer to the robot with a lower available energy (b).

at the meeting location with a non-negative battery level. The construction of energy-optimal motions that realize the solution to the first sub-problem becomes the second sub-problem that will be addressed in Section 5.2. We start with a formulation of the first sub-problem and later present its solution in Section 5.3. In Section 5.4 we present a robotic implementation of the resulting algorithm to show that this is viable in practice.

Suppose we have N planar robots, whose task it is to meet at a common location. Let $p_i \in \mathbb{R}^2,\ i = 1, \ldots, N$, denote the opinions of the individual robots as to where this location should be. Similarly, let $\tau_i \in \mathbb{R}_+$ be the time at which robot i thinks the team should meet. Since the robots should agree on when and where to meet, we would like to determine a distributed protocol that makes $p_i = p_j$ and $\tau_i = \tau_j,\ \forall i, j \in \{1, ..., N\}$.

Now, what makes this a non-standard consensus problem is that the robots need to meet without draining their batteries and we would moreover like to achieve rendezvous quickly. Thus, the corresponding, constrained optimization problem becomes

Energy-Constrained Spatio-Temporal Rendezvous

$$\min_{(p,\tau)} \sum_{i=1}^{N} \left(\frac{1}{2}\rho\tau_i^2 + \left(\sum_{j \in N_i} \|p_i - p_j\|^2 + \|\tau_i - \tau_j\|^2 \right) \right)$$

$$\text{s.t. } h_i(p_i, \tau_i) \leq 0,\ \forall i = 1, ..., N, \tag{51}$$

where $p = [p_1^T, \ldots, p_N^T]^T,\ \tau = [\tau_1, \ldots, \tau_N]^T$, and N_i is the set representing robot i's neighborhood – the robots with whom robot i can share information. Finally, $h_i(p_i, \tau_i) \leq 0$ is

the constraint (to be determined) that ensures that robot i can indeed reach p_i in time τ_i without running out of energy. And even though the robots are assumed to be similar, they will start at different positions and with different initial battery levels, which we encode through the subscript i in the individual constraints.

The first term in the cost ensures that the robots meet quickly, where the weighting factor $\rho > 0$ determines how strongly this term affects the cost. The second term ensures that the decision variables, p and τ, across all robots get "close" in the least-squares sense. As such, what we consider is not exact rendezvous, but rather that the robots end up sufficiently close together at roughly the same time.

In order to find a mathematical expression for $h_i(p_i, \tau_i)$, we need to describe *how* the robots are moving and how that movement in turn affects the battery life of the robots. To this end, decisions must be made about the dynamics of the robots and their energy-consumption model.

5.2 Single Robot Control for Minimum Energy Consumption

The coupling between the different robot movements arises from the cost associated with the spatio-temporal rendezvous problem. However, the battery constraint is not coupled, that is robot i's battery level dynamics does not depend on robot j's battery levels. As such, in order to arrive at the constraint $h_i(p_i, \tau_i)$, it is sufficient to consider a single agent in isolation, which will be the case in this section.

The pose of a differential-drive mobile robot comprises of position and orientation. If we let (q_{i1}, q_{i2}) be the position of robot i, moving in the direction q_{i3}, the kinematics become

$$\begin{aligned} \dot{q}_{i1} &= v_i \cos q_{i3} \\ \dot{q}_{i2} &= v_i \sin q_{i3} \\ \dot{q}_{i3} &= \omega_i. \end{aligned}$$

For the purpose of analysis, we will assume that the robots move along straight lines, i.e., $\omega_i = 0$. We justify this assumption by using the fact that the shortest distance path between two planar points is the straight-line path between them and since we wish to minimize

energy consumption due to mobility, we find it reasonable to have each robot travel the shortest distance possible in order to reach its destination.

As a consequence, we can focus on this one-dimensional motion, and let the corresponding robot position be x_{i1}, where $\dot{x}_{i1} = v_i$. However, to capture battery usage, we need to go beyond kinematic models, and we let $x_{i2} = v_i$, with dynamics $\dot{x}_{i2} = u_i$, with u_i being the input, thus connecting to the rich literature on double integrator coordination, e.g., [90–93].

Finally, if we let x_{i3} denote the available battery level, the dissipation of energy is given by quadratic functions of the velocity and acceleration of the robot, similar to the model used in [94], yielding the combined dynamics

$$\begin{aligned} \dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= u_i \\ \dot{x}_{i3} &= -x_{i2}^2 - \alpha u_i^2, \end{aligned} \tag{52}$$

where $\alpha > 0$ determines how strongly acceleration affects the dissipation of energy in comparison to the effect of the velocity. Clearly, the battery level is decreasing while the robot is moving, hence we need that $\dot{x}_{i3} \leq 0$ at all times, which is satisfied by these dynamics. Also, if the robot is not accelerating, but still moving (i.e. has non-zero velocity), the battery level should still be decreasing.

As the ambition is to move in a manner that conserves energy, we seek to find the control input u_i that minimizes the total energy consumed, $x_{i3}(0) - x_{i3}(T)$. This optimal control problem can be formulated as

$$\min_{u_i} \int_0^T -\dot{x}_{i3}(t)\,\mathrm{d}t$$

subject to the constraints that the robot starts and ends at rest and reaches its target position while ensuring that the battery level is non-negative at the end of the maneuver, i.e., such

that

$$\begin{aligned} x_{i1}(0) &= x_{i10} & x_{i1}(T) &= x_{i1T} \\ x_{i2}(0) &= 0 & x_{i2}(T) &= 0 \\ x_{i3}(0) &= x_{i30} > 0 & x_{i3}(T) &\geq 0. \end{aligned}$$

Here T is the time over which the maneuver is defined – later to be made an explicit part of the problem – and x_{i10}, x_{i30} are the given initial positions and battery levels, respectively, and x_{i1T} is the given final position. It should be noted that this problem may not have a solution, e.g., if T is too small, and in subsequent paragraphs we provide the solution to this optimal control problem as well as characterize when said solution exists.

Therefore, in order to solve this problem analytically, we will remove the constraint that the final battery level be non-negative, which allows us to remove x_{i3} as a state in the optimal control design altogether. We will later re-incorporate x_{i3} and choose T in order to ensure that the final battery level is non-negative, thus characterizing when a solution to the original problem exists. This relaxed problem that we first solve is

Single Agent Energy Minimization

$$\min_{u_i} \int_0^T (x_{i2}^2(t) + \alpha u_i^2(t))\, \mathrm{d}t \tag{53}$$

subject to the dynamics

$$\begin{aligned} \dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= u_i \end{aligned} \tag{54}$$

and the fixed initial and final conditions

$$\begin{aligned} x_{i1}(0) &= x_{i10} & x_{i1}(T) &= x_{i1T} \\ x_{i2}(0) &= 0 & x_{i2}(T) &= 0. \end{aligned} \tag{55}$$

This optimal control problem is solved for u_i, which is given in the following theorem.

Theorem 11. *The control signal u_i that solves the problem in* (53)*, subject to the dynamics in* (54) *and to the boundary conditions in* (55)*, is given by:*

$$u_i = -\frac{1}{2\alpha}\left(c_{i1}\mathrm{e}^{\frac{1}{\sqrt{\alpha}}t} + c_{i2}\mathrm{e}^{-\frac{1}{\sqrt{\alpha}}t}\right), \tag{56}$$

where

$$c_{i1} = \frac{1}{2}\mathrm{e}^{-\frac{1}{\sqrt{\alpha}}T}\left(\nu_{i2} - \nu_{i1}\sqrt{\alpha}\right)$$

and

$$c_{i2} = \frac{1}{2}\mathrm{e}^{\frac{1}{\sqrt{\alpha}}T}\left(\nu_{i2} + \nu_{i1}\sqrt{\alpha}\right),$$

with

$$\nu_{i1} = \frac{2(x_{i10} - x_{i1T})}{T - 2\sqrt{\alpha}\tanh\left(\frac{T}{2\sqrt{\alpha}}\right)}$$

and

$$\nu_{i2} = -\sqrt{\alpha}\tanh\left(\frac{T}{2\sqrt{\alpha}}\right)\nu_{i1}.$$

Proof. Following the Pontryagin Maximum Principle for fixed-endpoint control problems, given, e.g., in [95], the Hamiltonian associated with the cost in (53) and dynamics in (54) is

$$H = x_{i2}^2 + \alpha u_i^2 + \lambda_{i1}x_{i2} + \lambda_{i2}u_i,$$

where the costates $(\lambda_{i1}, \lambda_{i2})$ satisfy

$$\dot{\lambda}_{i1} = -\frac{\partial H}{\partial x_{i1}} = 0$$
$$\dot{\lambda}_{i2} = -\frac{\partial H}{\partial x_{i2}} = -2x_{i2} - \lambda_{i1}.$$

We let $\lambda_{i1} = \nu_{i1}$, for some constant ν_{i1} on the entire interval since x_{i1} is determined at time T. This yields $\dot{\lambda}_{i2}(t) = -2x_{i2}(t) - \nu_{i1}$ and we set $\lambda_{i2}(T) = \nu_{i2}$ for some constant ν_{i2}, leaving us with the two unknowns, ν_{i1} and ν_{i2}.

The optimality condition on u_i is

$$\frac{\partial H}{\partial u_i} = 2\alpha u_i + \lambda_{i2} = 0 \;\Rightarrow\; u_i = -\frac{\lambda_{i2}}{2\alpha} \tag{57}$$

and the second costate λ_{i2} can be found by differentiating $\dot{\lambda}_{i2}$, plugging in u_i from (57) and then solving the resulting second-order differential equation

$$\ddot{\lambda}_{i2} = \frac{1}{\alpha}\lambda_{i2},$$

with boundary conditions $\lambda_{i2}(T) = \nu_{i2}$ and $\dot{\lambda}_{i2}(0) = \dot{\lambda}_{i2}(T) = -\nu_{i1}$ due to the boundary conditions on x_{i2}. The solution is

$$\lambda_{i2} = c_{i1}\mathrm{e}^{\frac{1}{\sqrt{\alpha}}t} + c_{i2}\mathrm{e}^{-\frac{1}{\sqrt{\alpha}}t}, \tag{58}$$

where

$$c_{i1} = \frac{\mathrm{e}^{-\frac{1}{\sqrt{\alpha}}T}}{2}\left(\nu_{i2} - \nu_{i1}\sqrt{\alpha}\right) \tag{59}$$

$$c_{i2} = \frac{\mathrm{e}^{\frac{1}{\sqrt{\alpha}}T}}{2}\left(\nu_{i2} + \nu_{i1}\sqrt{\alpha}\right). \tag{60}$$

By plugging (58) into the equation for u_i in (57), u_i is completely determined. And since $\ddot{x}_{i1} = u_i$, we can integrate u_i twice to obtain

$$x_{i1} = -\frac{1}{2}\left(c_{i1}\mathrm{e}^{\frac{1}{\sqrt{\alpha}}t} + c_{i2}\mathrm{e}^{-\frac{1}{\sqrt{\alpha}}t} + \nu_{i1}t - c_{i1} - c_{i2}\right) + x_{i10}, \tag{61}$$

with terminal constraint $x_{i1}(T) = x_{i1T}$. By substituting the expressions for c_1 and c_2 from (59) and (60) into (61) and using the terminal constraint, we get

$$\nu_{i1}\left(T - \sqrt{\alpha}\sinh\left(\frac{T}{\sqrt{\alpha}}\right)\right) + \nu_{i2}\left(1 - \cosh\left(\frac{T}{\sqrt{\alpha}}\right)\right) = 2\left(x_{i10} - x_{i1T}\right), \tag{62}$$

which, together with (59) and (60) and the condition $\dot{\lambda}_{i2}(0) = -\nu_{i1}$, gives

$$\nu_{i2} = -\sqrt{\alpha}\tanh\left(\frac{T}{2\sqrt{\alpha}}\right)\nu_{i1}, \tag{63}$$

and

$$\nu_{i1} = \frac{2(x_{i10} - x_{i1T})}{T - 2\sqrt{\alpha}\tanh\left(\frac{T}{2\sqrt{\alpha}}\right)}, \tag{64}$$

where everything on the right hand side of (64) is known. Hence we know ν_{i2}, and can compute c_{i1} and c_{i2}, which describe λ_{i2} completely, and thus u_i is also completely determined. ■

A direct consequence of Theorem 11 is that given T, x_{i10}, and x_{i1T}, the optimal control input that minimizes the total energy consumed throughout the move can be explicitly found. It should be noted, however, that the initial battery level does not matter to the control input and, as a consequence, we may indeed end up with a negative final battery level, which is impractical. But, the optimal control construction can be used to drive the system *until* the battery level is completely drained (i.e., $x_{i3} = 0$), which in turn corresponds to the shortest amount of time, $T_{min,i}$, in which the move can be completed such that the final battery level is non-negative. Moreover, $T_{min,i}$ certainly depends on the initial battery level, x_{i30}, even if u_i does not.

In order to determine the shortest amount of time in which the motion can be completed, an expression for the battery level $x_{i3}(t)$ is needed. It is found by substituting the optimal control input u_i from (56) and the corresponding velocity x_{i2} into the dynamics for x_{i3}, given in (52), and integrating. $T_{min,i}$ is then derived by computing $x_{i3}(T_{min,i})$ and setting this expression equal to zero, i.e.,

$$\frac{(x_{i10} - x_{i1T})^2}{T_{min,i} - 2\sqrt{\alpha}\tanh\left(\frac{T_{min,i}}{2\sqrt{\alpha}}\right)} - x_{i30} = 0. \tag{65}$$

Although this expression is quite cumbersome to solve explicitly, numerical solutions are easy to come by, relating the minimum time in which the robot can achieve the total displacement, $|x_{i1T} - x_{i10}|$, with the initial battery level, x_{i30}. As seen in Figure 20, $T_{min,i}$ increases with increasing distance traveled, as is to be expected. This is intuitive because in order to travel a further distance using the same amount of energy, the robot must travel slower, therefore taking longer. Also, a robot with a higher initial battery level can travel a specified distance in less time than a robot with a lower initial battery level since a higher initial battery level means that the robot is able to travel faster.

If we relax the constraint that the final battery level must be zero and instead let it be greater than or equal to zero, we get a similar condition on the initial battery level

$$x_{i30} \geq \frac{(x_{i10} - x_{i1T})^2}{T - 2\sqrt{\alpha}\tanh\left(\frac{T}{2\sqrt{\alpha}}\right)}, \tag{66}$$

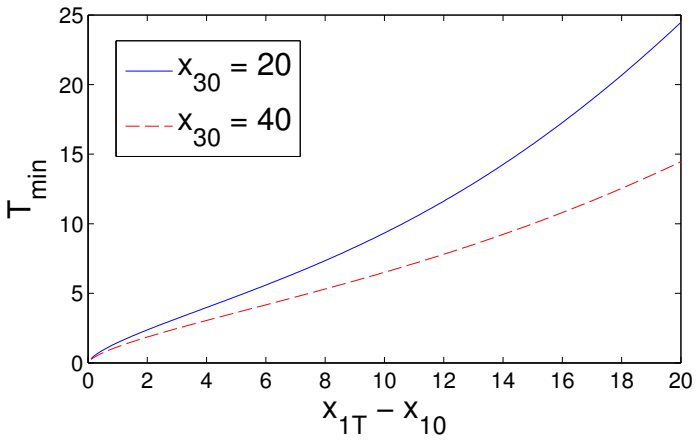


Figure 20. Depicted are curves that represent the minimum time (T_{min}) required for a robot to travel a specified distance ($x_{1T} - x_{10}$), given that the robot has an initial battery level of 20 (solid), and an initial battery level of 40 (dashed).

where the expression on the right-hand side is the amount of energy lost during the motion, which must be less than or equal to the initial battery level in order to have a non-negative battery level at time T. This inequality constraint is exactly what we set out to find, and, by recalling that (q_{i1}, q_{i2}) is the two-dimensional position of robot i, p_i is robot i's opinion of the location to meet, and τ_i is the time at which robot i thinks the team should meet, we can connect this back to the two-dimensional problem by replacing $(x_{i10} - x_{i1T})^2$ with $\|(q_{i1}(0), q_{i2}(0))^T - p_i\|^2$ and T with τ_i. A slight rearrangement of terms gives the sought-after constraint for the spatio-temporal rendezvous problem from the previous section,

$$h_i(p_i, \tau_i) = \|(q_{i1}(0), q_{i2}(0))^T - p_i\|^2 - (x_{i30}) \left[\tau_i - 2\sqrt{\alpha} \tanh\left(\frac{\tau_i}{2\sqrt{\alpha}} \right) \right].$$

5.3 Multi-Agent Energy-Aware Coordination Algorithm

Now that we have an expression for the constraint, h_i, we can tackle the energy-constrained, spatio-temporal rendezvous problem in (51). This is an optimization problem with a global cost function and local constraints, since all robots are minimizing the same objective function, but each robot has its own constraint that only depends on its own decision variables, i.e. p_i and τ_i.

Since gradient-descent methods are particularly well-suited for distributed implementations, this problem can be solved via the primal-dual gradient laws for constrained optimization, as described in [96], using the update dynamics given by

$$\begin{aligned}
\dot{p}_i &= -\sum_{j\in N_i}\Big(p_i - p_j\Big) - \frac{\partial h_i}{\partial p_i}(p_i,\tau_i)\,\mu_i \\
\dot{\tau}_i &= -\sum_{j\in N_i}\Big(\tau_i - \tau_j\Big) - \rho\tau_i - \frac{\partial h_i}{\partial \tau_i}(p_i,\tau_i)\,\mu_i \\
\dot{\mu}_i &= \begin{cases} h_i(p_i,\tau_i) & \text{if } h_i(p_i,\tau_i) > 0 \text{ or } \mu_i > 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{67}$$

where the dynamics associated with the Lagrange multiplier, μ_i, ensure that the multipliers remain positive, which is necessary because of the Karush-Kuhn-Tucker conditions associated with the inequality constraints. Note that this is a decentralized algorithm since each robot needs only its own p_i, τ_i, and μ_i values and the p_j and τ_j values of its neighbors. Each robot's μ_i dynamics only depend on its own values of p_i, τ_i, and μ_i, and thus these multipliers do not have to be shared among neighboring robots.

This algorithm was implemented in MATLAB for a network of three simulated robots with one-dimensional positions. The robots' initial positions are 10, 20, and 30, with initial battery levels of 30, 30, and 5, respectively. The initial meeting point for robot i, i.e., p_i, was set to robot i's initial position, x_{i10}, to indicate that robot i would initially like to meet the other robots at its own starting position. Each robot's initial meeting time, τ_i, was set to 100, so that the energy constraint ($h_i(p_i,\tau_i) \leq 0$) would be satisfied initially for all i.

The dynamics in (67) were executed in simulation using an ODE solver with a variable step Runge-Kutta method to obtain approximate solutions for p_i, τ_i, and μ_i, for $i = 1, 2, 3$. Figure 21 shows the evolution of the meeting points, p_i, and Figure 22 shows the evolution of the meeting times, τ_i. In this example, the robots do indeed end up agreeing to meet closer to the robot with the lowest initial battery level, as was to be expected. In this simulation, the final meeting point values were $p_1 = 24.1925$, $p_2 = 24.2002$, and $p_3 = 24.2079$, whereas the final τ values were $\tau_1 = 11.1250$, $\tau_2 = 11.1169$, and $\tau_3 = 11.1203$.

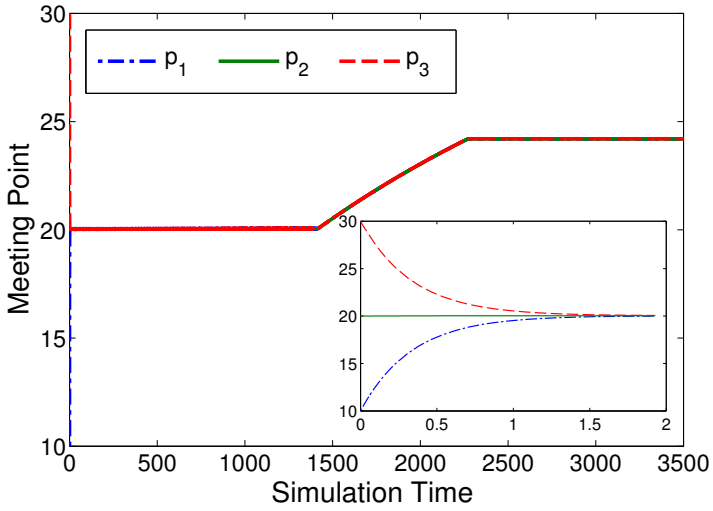


Figure 21. Depicted are the desired meeting positions, p_i, as a function of simulation time. The initial meeting position for each robot is that robot's initial position (here the initial positions are 10, 20, and 30). The zoomed in portion shows the p_i values for simulation time 0 to 2.

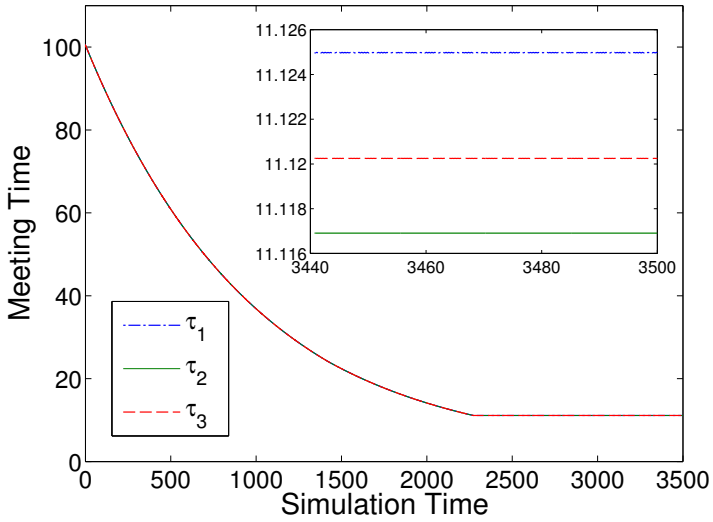


Figure 22. Depicted are the curves of the times, τ_i, in which each robot would like to meet the other robots, where 100 was used as the initial meeting time for all robots. The zoomed in portion shows the τ_i values for the three robots for simulation time 3440 to 3500.

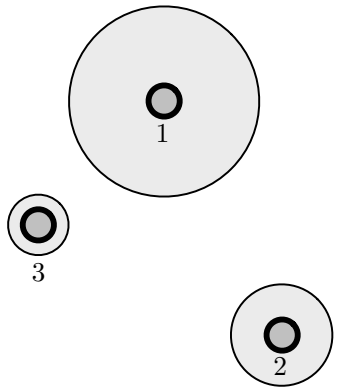


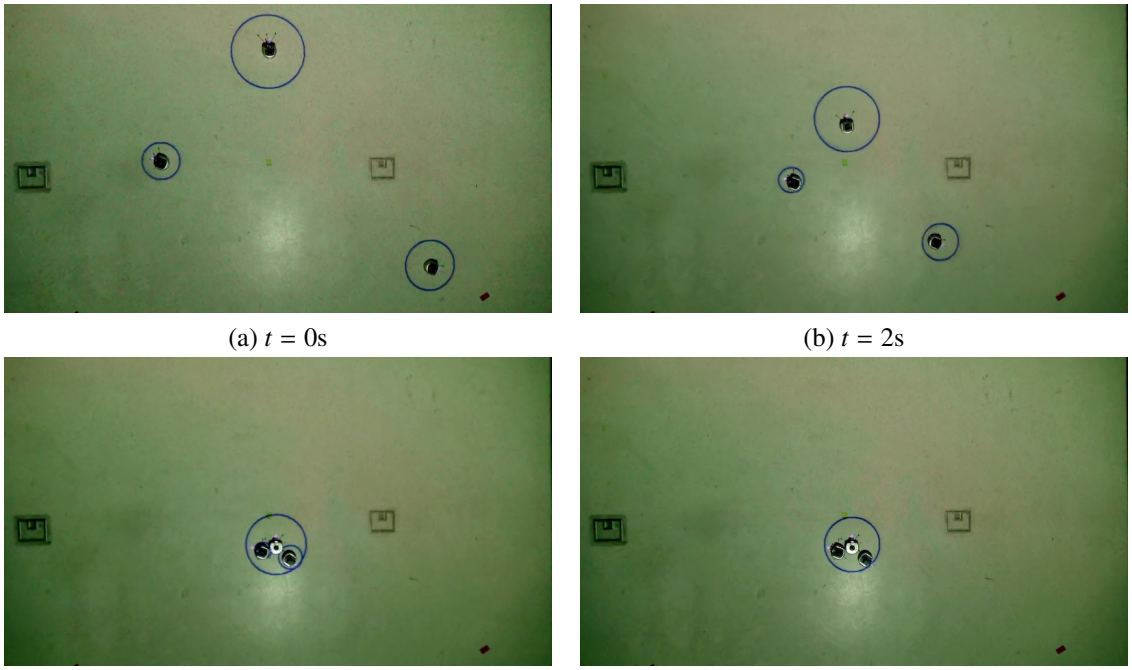
Figure 23. Initial positions of the robots, where the radius of the circle around each robot represents its initial battery level.

5.4 Implementation on a Robot Team

This algorithm was implemented on a team of Khepera III differential-drive mobile robots, each having a 600MHz ARM processor with 128MB RAM, embedded Linux, and a wireless card for enabling communication over a wireless router. Ten Optitrack motion capture cameras were used to obtain highly accurate position and orientation data for the robots, providing the information required for the algorithmic implementation and to project the visualization of the robot battery levels onto the floor.

We used three robots, with initial positions given by $[0, -1]$, $[-1.5, 1]$, and $[1, 0]$, and initial battery levels given by 2, 1, and 0.5, respectively. The robots' initial positions and relative initial battery levels are depicted in Figure 23, where the radii of the circles surrounding the robots are pictorial representations of the available battery levels. The meeting points and meeting times were obtained by solving the constrained optimization problem in a distributed manner, with solution $p_1 = [-0.0304, 0.4278]$, $p_2 = [-0.0318, 0.4283]$, $p_3 = [-0.0289, 0.4272]$, and $\tau_1 = 6.4886$, $\tau_2 = 6.4890$, $\tau_3 = 6.4889$. These decision variables were computed before the robots started moving.

The corresponding, optimal trajectories were then executed on the robots by having them travel along straight-line paths towards the meeting point with the corresponding, optimal linear velocities. The robots first turn in place until they are positioned in such



(a) $t = 0$s (b) $t = 2$s

(c) $t = 4$s (d) $t = 6$s

Figure 24. Snapshots taken over time of robots running energy-constrained rendezvous. The circles, projected on top of the robots using a projector, represent the remaining battery levels of the robots. The video can be found online at https://youtu.be/KNUnwAI3Lec.

a way that they can reach their goal position by traveling along straight-line paths. This initial turning is deemed negligible in terms of energy consumption. While the robots are driving to the rendezvous point, the modeled battery life of each is depicted by projected circles on top of each robot, where the radius of each circle's radius is proportional to the respective robot's remaining battery level.

Four snapshots of the robots driving to the meeting point can be seen in Figure 24. It can be seen that at the end of the move, one of the robots has significant battery life remaining, while the other two robots have depleted their batteries. It can also be seen that instead of the robots meeting at the average of their initial positions ($[-0.1667, 0]^T$), as would happen if they were running the standard consensus protocol, they instead meet closer to the robots with the lower initial battery levels, i.e. robots 2 and 3. The difference between the average of their initial positions and where they actually meet can be seen in Figure 25. The modeled battery level trajectories are shown in Figure 26.

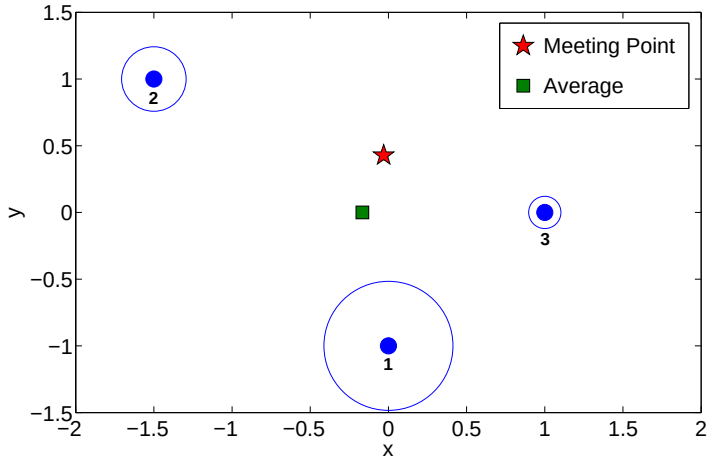


Figure 25. Depicted are the initial positions of the robots, along with the average of these initial positions (square) and the meeting point that they decided upon using the algorithm derived in this chapter (star). The robots' respective initial battery levels are given by circles around them.

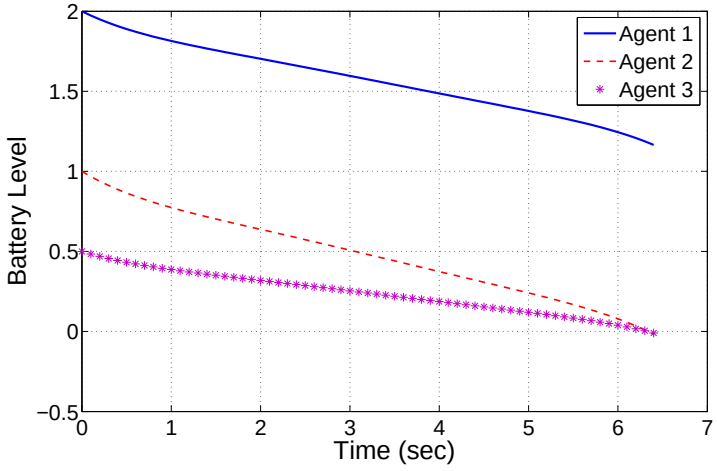


Figure 26. Depicted are the modeled battery life trajectories for each robot over the time duration of the experiment. As can be seen, robots 2 and 3 finish the experiment with battery levels that are close to zero, while robot 1 finishes with a positive battery level.

5.5 Conclusions

In this chapter, we presented an energy-constrained strategy for allowing a network of robots to achieve rendezvous in the shortest possible amount of time without depleting their batteries. We formulated the problem of finding the best location for the robots to meet given their initial positions and battery levels as a constrained optimization problem. The energy constraints arise by solving the problem of how to control a single robot that is moving from some given initial position to some desired final position such that the overall energy consumption, due to mobility is minimized. This allows the robots to achieve rendezvous as quickly as possible while also minimizing the amount of energy that each robot consumes throughout the motion. The algorithms described in this chapter were implemented on a team of Khepera III mobile robots to show that this is viable in practice.

CHAPTER 6

HAPTIC HUMAN-SWARM INTERACTIONS

While the majority of this thesis focused on understanding human-robot teams through rigorous theoretical analysis, this chapter presents an experimental approach. The objective of this work is to design useful human-swarm interactions by investigating different ways to feed information about the state of the robot swarm back to the human, or so-called operator. Haptics was used as the method by which to relay the information and manipulability is the metric that was fed to the operator, which tells the operator how effectively he or she is controlling the so-called "leader" robots. User studies were done to find the best mapping between the manipulability index and the feedback force applied to the haptic device that was being used by the operator. The work in this chapter was published in [97] and [15].

6.1 Background

The current approach to operating autonomous vehicles is one-to-one, i.e., a single operator is interacting with a single vehicle, or even many-to-one, where multiple operators are needed to control a single vehicle, as is for example the case when controlling unmanned drones. However, in order to invert this many-to-one relationship, i.e., to enable single operators to control and interact with multiple vehicles, new interaction abstractions are needed.

Some initial work on human-swarm interactions has been done, where different interaction abstractions have been proposed. For example, [98] investigated how user interfaces should be structured in order to facilitate control of multi-robot teams. A related study focuses on the design of useful displays that provide sufficient situational awareness without overloading the operator with data, e.g., [99]. There has also been work being done on developing useful abstractions for human-swarm interactions. In [100], so-called motor schema are defined and the human operator acts on the team as an additional motor schema. In a similar manner, in [101], strengths of biologically inspired entities are influenced by

the user – other similar approaches can for example be found in [102]. In [103], a human controls the behavior of a group of robots by specifying areas of importance through the use of a touch-sensing tablet.

However, in order to employ haptic interaction modalities for a team of robots, what is needed is a direct way of mapping the injected control signals onto a force that is experienced by the operator. But, as has been observed repeatedly in the literature, e.g., [104, 105], when interacting with multi-robot teams, the organization of the interaction dynamics matters. In other words, if the individual robots are nodes in a graph, interactions between pairs of robots can be encoded through edges between the corresponding nodes in the graph. The resulting graph structure is known as the interaction network [47], and its topology, i.e., what the graph looks like, has a direct impact on how easy or hard it is to control the network, as shown in [105]. As a result, the mapping from control inputs to haptic feedback must take the underlying network topology into account.

Manipulability is a standard notion in robotics for describing how effectively joint angle velocities translate into end-effector velocities for high degree-of-freedom manipulators [106–108]. In [109, 110], this idea was moved to the multi-robot domain in the context of "leader-follower" control. In this setup, a subset of the robots are leader robots, whose velocities can be controlled directly (corresponding to the manipulator joints), while the remaining agents are the followers (corresponding to the end-effector). In this work, we pursue this idea as a generator of haptic forces experienced by the operator, i.e., high manipulability (swarm is easy to control) yields small forces and low manipulability (swarm is hard to control) yields large forces.

The idea of using haptics to facilitate human interactions with multi-robot teams has been investigated previously. For example, [111] used haptic feedback to relay information regarding the presence of obstacles and other external disturbances to the operator via a device that was simultaneously being used to control the movement of a group of UAVs. There are other examples where a haptic device has been used to control a group of robots

as well as provide feedback to the user. For example, [112–114], explore bilateral teleoperation of groups of robots by a single master in the presence of communication delays. Similarly, in [115], a human operator teleoperates a robot team while also controlling the degree of connectivity and haptic feedback is used to inform the operator of the discrepancy between the desired minimum degree of connectivity and the value implemented by the control action. In [116], the effectiveness of haptic feedback to an operator controlling a robotic swarm is explored through user experiments.

The work we present in this chapter differs in that we investigate the effectiveness of different mappings between a team-level property, namely manipulability, and the haptic feedback force that the operator experiences when interacting with a select subset of robots in the team, whereas the work done in the aforementioned literature picks one such mapping a priori to relay information to the operator. It also differs in the fact that we are using manipulability to let the human operator know how effectively the leader of the group is controlling the motion of the followers.

The outline of the remainder of this chapter is as follows: Section 6.2 provides a characterization of swarm-level properties that are appropriate for haptic feedback. In Section 6.3, we introduce the idea of leader-follower networks of mobile robots and discuss how the notion of a manipulability index can be applied to these networks under a general choice of interaction dynamics. In Section 6.4, we justify the use of haptics to relay this manipulability information to a human operator that is tasked with controlling a multi-robot team and discuss the different types of mappings between manipulability and haptic force that were explored. In Section 6.5, the setup of the user experiments are described and the results comparing these different mappings are presented. In Section 6.6, we draw concluding remarks.

6.2 Haptic Swarm Control

When a human operator is controlling a swarm by providing inputs for the team, it would be useful for the operator to be made aware of certain swarm-level properties that it can

use to make decisions. For example, if the robot swarm is going to run into an obstacle, the operator would want to know so that it can direct the swarm away from the obstacle. Haptic technology provides a way in which these properties can be relayed to the user via force feedback, while he or she is controlling the motion of the swarm. We investigate here what constitutes a haptic-appropriate swarm-level property and what is needed to turn such a property into useful haptic forces.

For the purpose of this discussion, we assume that the human operator is controlling the velocity of the leader of the swarm and the remaining agents, or the followers, are tasked with maintaining pairwise inter-robot distances, as is standard in much of the multi-robot literature, e.g., [8, 47]. To illustrate, if agents i and j are adjacent in the information-exchange network, they are tasked with maintaining the distance between them, $\|x_i - x_j\|$, to a desired, pre-specified, positive value d_{ij}. If a follower is adjacent to the leader, only the follower's dynamics will strive to maintain the distance between the two agents. This type of network is known as "leader-follower". The operator controls the leader's velocity through the use of a haptic device, which is also used to generate the feedback forces that relay the swarm-level information. This is desirable because only one device is being used by the operator and it eliminates the need for any intermediary senses. The operator should be able to apply the information given by the haptic device without having to think much about it.

We first investigate what characteristics are needed by a swarm-level property for it to be an appropriate haptic feedback signal in this setting. It has been shown, for example, in [117] and [118], that when haptic delays are present, the person using the haptic device perceives the force feedback to be weaker than it is in actuality. In order for the operator to feel the forces with the strength that they were intended to have, the delay caused by the computation of the force feedback should be minimized. In order to minimize delay, the swarm-level property used for haptic feedback should be an instantaneous notion. That is, it should address instantaneous effects that the input, given by the human operator, has on

the swarm of agents. This way, as soon as the operator changes the state of the swarm by moving the haptic device, a new haptic feedback force can be computed instantaneously, giving the operator instant feedback about whether that input motion was "good" or "bad".

All haptic devices are limited by the amount of force they can produce, so we need to map the value of the swarm-level property to an appropriate amount of force in the range that the device can produce. In order to generate forces that are easily distinguishable by the operator, it seems desirable to map the full range of the swarm-level property to the full range of the haptic device. In order to do so, the swarm-level property needs to have both a maximum and minimum value, known a priori so that the mapping can be be defined ahead of time and remain constant throughout the human-swarm interaction task. The swarm-level property should also be continuous as to not cause discontinuities in the haptic feedback force and so that the mapping from the property to the feedback force is straight-forward.

In order for the haptic feedback to be useful, the operator needs to know *how* to use the information being relayed to him or her. Therefore, the properties being used for haptic feedback should also be beneficial to the user in completing the task at hand. For example, haptic feedback indicating obstacles in an environment would be useful to an operator who is tasked with moving a swarm through an environment without colliding with obstacles. If the task were different, this type of feedback may not be as useful.

In addition to being useful, the haptic feedback needs to be forceful enough so that it can successfully influence the user's decisions. If we wish to impede the user from moving the leader of the swarm in a certain direction, the force needs to be strong enough to overcome the force that the operator is applying to the device, or at the very least be strong enough for the operator to notice the resistance. Device limitations aside, this is a matter of picking an appropriate mapping between the swarm-level property and the haptic force.

If the properties discussed in this section are met, haptics can be effectively used to assist a human operator in controlling a swarm of mobile agents, by allowing the operator

to be informed about the state of the swarm as a whole. In the next section we will give an example of a swarm-level property that fits our needs.

6.3 Manipulability and Leader-Follower Control

For this work, we use the leader-follower approach to controlling teams of mobile robots, whereby a subset of the robots (the leaders) are controlled directly, and the control signals are indirectly propagated through the network through the leaders' motions. Since our aim is to provide meaningful and effective haptic feedback to the operator, an instantaneous notion is needed for how easy or hard it is to interact with the robot team. This means that the standard notion of point-to-point controllability may not be ideal since it is a notion that concerns itself with the (possibly long-term) transfer of the system from one state to another. In contrast to controllability, which is not an instantaneous property, manipulability is a promising candidate for providing the needed, instantaneous, haptic feedback. It is a term borrowed from the robotic manipulation literature (e.g., [106–108]), and in this section we recall the key manipulability ideas.

Consider a network consisting of N robots, divided into groups of leaders and followers, such that there are N_f followers and N_l leaders, with $N_f + N_l = N$. Assume that, at time t, each robot is located at position $x_i(t) \in \mathbb{R}^d,\ i = 1, \ldots, N$, where d is the spatial dimension under consideration, e.g., $d = 2$ in the case of planar robots, $d = 3$ if they move in a three dimensional space, and so forth. We can aggregate the positions together to describe the overall position of the robot team at time t, which is given by $x(t) = [x_1^T(t), ..., x_N^T(t)]^T \in \mathbb{R}^{Nd}$. For the sake of notational simplicity, we assume that the indexing of the agents is such that the first N_f agents are the followers, and the last N_l agents are the leaders. Under this indexing, we have that $x(t) = [x_f^T(t), x_l^T(t)]^T$, where $x_f(t) = [x_1^T(t), ..., x_{N_f}^T(t)]^T \in \mathbb{R}^{N_f d}$ and $x_l(t) = [x_{N_f+1}^T(t), ..., x_N^T(t)]^T \in \mathbb{R}^{N_l d}$.

In a leader-follower network, the idea is to let the leaders' velocities be controlled by external inputs (the control signals provided by the operator), and then let the followers' velocities be defined through pairwise interactions between adjacent agents, as is quite

standard in the multi-robot literature, e.g., [8, 47]. As a number of different such pairwise interaction laws have been proposed, our ambition is to design a haptic interaction framework that is largely agnostic to the actual choice of interaction dynamics. However, some choices will have to be made, and we assume that the team of robots is tasked with maintaining desired, pairwise inter-robot distances. This is a rather general version of the so-called formation control problem, and what it means is simply that whenever robots i and j are adjacent in the underlying information-exchange network, their task is to make the distance between them, $\|x_i - x_j\|$, as close as possible to a desired distance d_{ij}. Not all robots necessary will be adjacent to all other robots, and thus the formation need not be rigid.

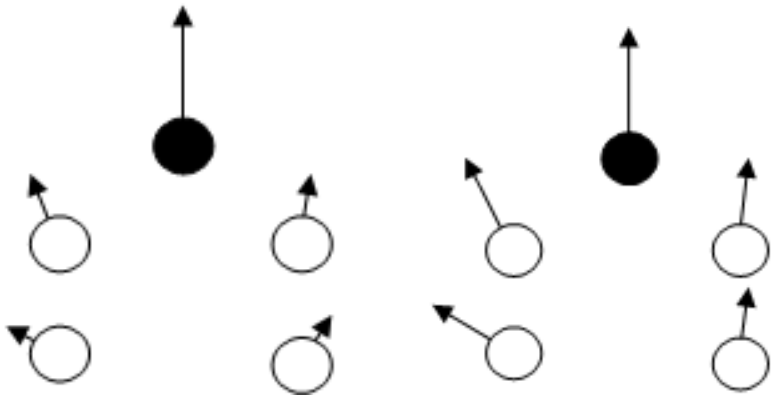
Using this formation-based leader-follower setup, one can formulate the multi-agent manipulability index in a manner that is immediately analogous to the way it is formulated for manipulators, namely as a ratio between the leaders' and the followers' velocities, i.e.,

Manipulability Index for Leader-Follower Networks

$$M = \frac{\|\dot{x}_f\|^2}{\|\dot{x}_l\|^2}. \tag{68}$$

An example of this idea is illustrated in Figure 27, where the multi-robot network on the left has a lower manipulability than the network on the right due to the fact that the leader-follower velocity ratio is smaller in network (a).

It is instructive to explicitly untangle what the manipulability index M in (68) actually depends on. Clearly it is a function of $\dot{x}_l$, since this is the control signal that is injected by the user. It moreover depends on the total configuration of the multi-robot network, i.e., where all the different robots actually are, x, as well as who they are adjacent to in the network, i.e., what robots are trying to maintain the desired distance to each other. If we let $V = \{v_1, \ldots, v_N\}$ denote the set of robots, we can define the set $E \subseteq V \times V$ as the unordered set of robot pairs between whom distances are maintained. That the set is unordered means



(a) Less Effective (b) More Effective

Figure 27. Effectiveness of interactions with a leader-follower multi-robot network based on manipulability ($N_l = 1$). The filled circle is the leader and the arrows represent the agents' velocities.

that $(v_i, v_j) \in E \Leftrightarrow (v_j, v_i) \in E$, i.e., if robot i cares about the distance to robot j, then robot j cares about the distance to robot i. Using the *vertex* set V together with the *edge* set E, we have actually specified the undirected graph $G = (V, E)$, that defines the information-exchange network in the multi-robot team. And, the manipulability index in (68) depends on this graph.

To summarize, we have that M in (68) depends at the very least on $\dot{x}_l, x$, and G. But, unfortunately, we need one more piece of information to be able to compute M, namely $\dot{x}_f$. And this quantity depends explicitly on the choice of interaction-dynamics. However, as our ambition is to be general and not over-design the haptic feedback to a particular choice of interaction dynamics, this obstruction must be remedied. In the manipulation literature, this problem does not arise, since the links in the manipulator are rigid, i.e., there is no interaction dynamics present in the way that it is present in a multi-robot network. In [110], it was explored how an assumption of rigidity on the links in the multi-robot network translated to a more general and easily computable, yet approximate, manipulability notion that does not depend on the interaction-dynamics, i.e.,

$$\tilde{M}(x, \dot{x}_l, G) \approx \frac{||\dot{x}_f||^2}{||\dot{x}_l||^2}.$$

For the sake of the clarity of the composition and for explicitly connecting to the haptic interaction modalities, we here recall the construction from [110].

To get at a rigid link approximation of what happens in the multi-robot network without having to explicitly specify the interaction dynamics, we assume that this dynamics is at least able to do what it was designed to do, i.e., get (close) to the desired inter-robot distances sufficiently fast. In other words, for the purpose of obtaining an approximate manipulability measure, we will assume that the desired distances $\{d_{ij}\}_{(v_i,v_j)\in E}$ are perfectly maintained by the followers at all times, i.e., $\|x_i(t) - x_j(t)\| = d_{ij},\ \forall (v_i, v_j) \in E,\ t \geq 0$. Note that unless the leaders move significantly faster than the followers, this approximation gives a reasonably good characterization of the team behavior under the influence of leader velocity inputs.

Under the rigid link approximation, the distance between connected robots are static, i.e. they do not change over time. If the trajectories of $x_i(t)$ are smooth and differentiable, then what this means is that

$$\frac{d}{dt}\|x_i(t) - x_j(t)\|^2 = 0,\ \forall (v_i, v_j) \in E,\ t \geq 0,$$

which expands to

$$(x_i - x_j)^T(\dot{x}_i - \dot{x}_j) = 0,\ \forall (v_i, v_j) \in E, \tag{69}$$

where we have suppressed the dependence on t for the sake of notational simplicity.

Using (69), the rigid-link approximation condition can be written in matrix form as

$$R(x)\dot{x} = 0,$$

where $R(x) \in \mathbb{R}^{|E|\times Nd}$ is the so-called *rigidity matrix* of the system, and where $|E|$ is the cardinality of the edge set. Or, if we explicitly call out the parts contributed by the leaders and the followers,

$$R(x, G)\begin{bmatrix} \dot{x}_f \\ \\ \dot{x}_l \end{bmatrix} = \Big[R_f(x, G)|R_l(x, G)\Big]\begin{bmatrix} \dot{x}_f \\ \\ \dot{x}_l \end{bmatrix} = 0,$$

where $R_f \in \mathbb{R}^{|E|\times N_f d}$ and $R_l \in \mathbb{R}^{|E|\times N_l d}$.

In [110] it was shown that this in turn implies that the follower velocities can be directly expressed as a function of the leader velocities (as well as x and G), as

$$\dot{x}_f = -R_f^{\dagger}(x, G)R_l(x, G)\dot{x}_l, \tag{70}$$

where $R_f^{\dagger}$ is the Moore-Penrose pseudoinverse of R_f. This relation gives the prescribed, approximate manipulability measure

Approximate Manipulability Index

$$\tilde{M}(x, \dot{x}_l, G) = \frac{\dot{x}_l^T J^T(x, G)J(x, G)\dot{x}_l}{\|\dot{x}_l\|^2}, \tag{71}$$

where $J(x, G) = -R_f^{\dagger}(x, G)R_l(x, G)$. And, returning to the discussion of what these measures should depend on, as well as a desire to not have to depend on the particular choices of interaction law, this approximate manipulability measure is what we will use as a generator of haptic feedback signals, which is the topic of the next section.

6.4 Haptic Manipulability

One consequence of the approximate manipulability index is that it is "easier" to move the team of robots in certain directions, and with certain choices of leaders. This observation needs to be formalized in order to map the manipulability index onto a meaningful haptic feedback signal. For example, as shown in [110], in the single leader case, the approximate manipulability takes a large value when the direction of the leader's motion coincides with that of the motion of the followers' centroid. In the remainder of this chapter, the focus will be on the single leader case, because the experiments consist of a single operator controlling a single leader robot, which is easy and intuitive for the operator to envision. It should be noted that we are not concerned with finding the maximum and minimum values of the approximate manipulability index, but rather with finding a mapping from manipulability to haptic feedback that can help a human operator effectively complete a multi-robot task.

Haptics are a natural choice for conveying manipulability because it requires no intermediary senses. The mechanism of control actuation is intimately tied to the feedback sensation generated by the haptic device. And, since the haptic device can also be used by the human operator to control the leader's velocity, this choice removes complexities that may arise from having two separate mechanisms for controlling the leader and conveying the manipulability information from the robot team to the operator. The main idea is that the operator should not have to think about how to apply the manipulability information, because the forces acting on the haptic device should force the operator away from directions that result in a lower manipulability.

In order to apply forces to the haptic device, a mapping between the manipulability of the network and the haptic force must be chosen. By choosing a force mapping that is a monotonically decreasing function of manipulability, the user is encouraged by lesser force to move the network in more manipulable directions, while being discouraged by greater force from moving in less manipulable directions. The goal is to encourage the user to move the system in directions of higher manipulability, so that the control input is more effective in terms of the response of the multi-robot team.

These mappings should moreover be constructed in such a way that the maximum force is returned to the operator when the multi-robot team produces zero manipulability in that direction. Similarly, the operator should feel the minimum amount of force when the input direction produces the greatest manipulability the system can achieve. And, it can be shown that the maximum approximate manipulability in a leader-follower network with a single leader is equal to the number of followers in the network, N_f.

In the single leader case with which we are concerned, i.e. $N_l = 1$, R_l can be expressed in terms of R_f as

$$R_l = -R_f \tilde{I}_f, \tag{72}$$

where $\tilde{I}_f = \mathbf{1}_{N_f} \otimes I_d$, where $\mathbf{1}_{N_f}$ is an N_f-dimensional column vector with 1s in all of its entries, $\otimes$ denotes the Kronecker product, and I_d denotes the $d \times d$ identity matrix. By

substituting this R_l into (70), we get

$$\dot{x}_f = -R_f^{\dagger} R_l \dot{x}_l = R_f^{\dagger} R_f (\mathbf{1}_{N_f} \otimes I_d) \dot{x}_l = R_f^{\dagger} R_f (\mathbf{1}_{N_f} \otimes \dot{x}_l).$$

Since $R_f^{\dagger} R_f$ is a projection matrix, we get $\|\dot{x}_f\|^2 \leq N_f \|\dot{x}_l\|^2$. Thus, the desired result,

$$\tilde{M} = \frac{\|\dot{x}_f\|^2}{\|\dot{x}_l\|^2} \leq N_f,$$

follows.

Hence, we know a priori what the maximum manipulability value can be. The minimum force exerted by the haptic device should be zero so that an operator moving the network in the most manipulable direction should not be encouraged by the haptic device to change directions.

We explore two possible classes of such mappings. These were chosen since they are, in a certain sense, canonical in that they recover different aspects of what constitutes a potentially useful mapping. One example of such a mapping is a linear function that inversely maps manipulability to haptic force,

$$F_{linear}(\tilde{M}) = H\left(1 - \frac{\tilde{M}}{N_f}\right),$$

where H is the maximum applicable force of the haptic device, $\tilde{M}$ is the approximate manipulability of the team, and (as before) N_f is the number of followers in the network.

This linear map does not encourage high manipulability in a particularly forceful way. And, it can be contrasted with an inverse exponential map,

$$F_{exponential}(\tilde{M}) = H\,\frac{e^{-\alpha\tilde{M}} - e^{-\alpha N_f}}{1 - e^{-\alpha N_f}}.$$

Here, α is a parameter that can be changed to adjust the rate of change of the force as a function of manipulability. In the next section, these choices are explored in an experimental setting, where users are tasked with solving a multi-robot task using both linear and exponential maps.

Figure 28. Initial configuration of robots and the two target locations, illustrated with black circles.

6.5 Experimental Procedures and Results

In order to analyze the effectiveness of different mappings between manipulability and haptic force, user experiments were performed. Ten subjects voluntarily participated in this study, where each subject was tasked with moving a leader-follower network of differential-drive Khepera III robots between different target locations. This choice of task was driven by the fact that in a number of multi-robot applications, team cohesion is provided by the local coordination and control laws, while high-level objectives, such as target locations or directions, are externally applied.

Each of the subjects controlled the velocity of the leader of the swarm by using a PHANTOM Omni haptic device, while manipulability information was relayed to the subject via feedback forces on the haptic device. During each run, the subject was required to direct the leader, and hence the robot team, via the haptic device, to one of the target locations and then to the other target location, in either order. The leader robot was to end up on top of each target location, which was marked with an 'X' on the floor. The initial configuration of the robots, along with the marked target locations, can be seen in Figure 28. The leader is the robot with a white styrofoam object on top of it.

In addition to the physical setup, there was a virtual environment that the subject could look at that showed the positions of the robots, the positions of the target locations, and the velocity of the leader (shown by an arrow with direction and magnitude). This was

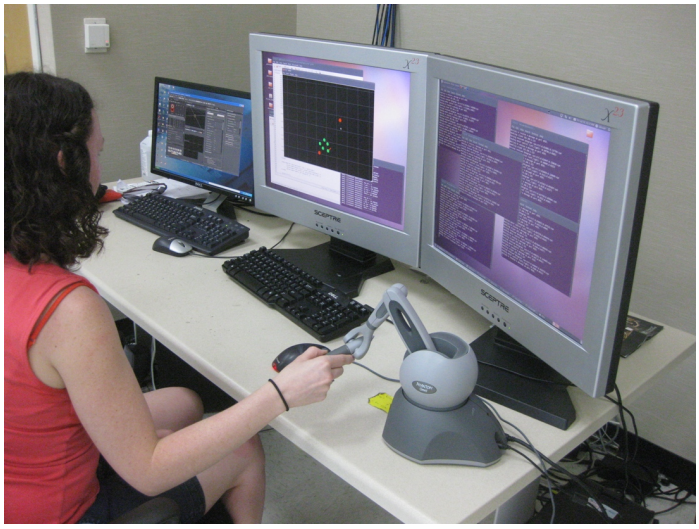


Figure 29. Photo of student utilizing haptic device while looking at the virtual environment (middle screen).

provided to give the subject a better feel for the environment. A photo of a subject using the haptic device while looking at the virtual environment can be seen in Figure 29.

The information-exchange network chosen for the user experiments was a line of four followers, with a single leader coming off of the middle two followers, forming a triangle. This configuration was chosen in order to allow for the network to not be rigid in the sense that it could fold and bend while respecting the desired inter-robot distances. This configuration is shown in Figure 30, where the lines between robots represent links that identify which robots can communicate with each other, or similarly, which robots are in each other's neighborhood set. The leader of the network is represented by a black circle. Each follower robot's control task is to maintain a desired distance between it and the other agents in its neighborhood set. These desired distances are the initial distances between agents, as seen in Figure 30. Numbering the follower robots in this figure from left to right, it is important to note that robots 1 and 4 each only have one robot in their neighborhood set, being robots 2 and 3, respectively. This means that the formation shown will not always be maintained, because robots 1 and 4 can move around a bit, as long as robot 1 maintains its distance to robot 2 and robot 3 maintains its distance to robot 4.

The mappings between manipulability and haptic force used in the user experiments

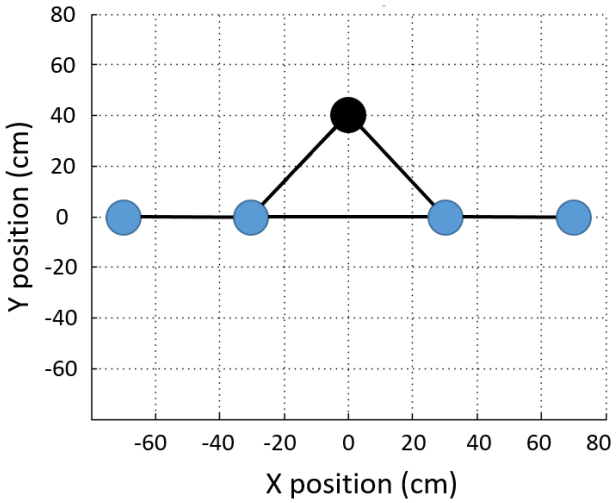


Figure 30. Initial configuration of the leader-follower network for the haptic human-swarm experiments.

consisted of a linear mapping and four exponential mappings with α parameters of 2, 0.5, -0.5, and -2. As discussed previously, these mappings are decreasing functions of manipulability, with a maximum value of one and a minimum value of zero. In order to encourage users to move the leader of the network in directions of highest manipulability, the maximum manipulability is mapped to zero haptic force and a manipulability of zero is mapped to a haptic force of one. The maximum value of manipulability is the same as the number of followers in the network, which is four in this case. The haptic force was applied in the opposite direction of the input velocity that was given by the human operator, so as to be a repulsive force that intends to impede motion in certain directions. These five mappings can be seen in Figure 31.

It should be pointed out that we did not explicitly test whether or not manipulability is indeed the best notion when interacting with multi-robot teams - both in terms of user experience and in terms of task completion rates. The main objective was to investigate different mappings from manipulability to haptic forces and to gauge their effects on the user experience. As such, the focus is on improving the human experience during the task so the operator can "feel" how easy or hard it is to move the team of robots as a whole. In this experiment, the task is to move the leader between target locations, and it is not a given

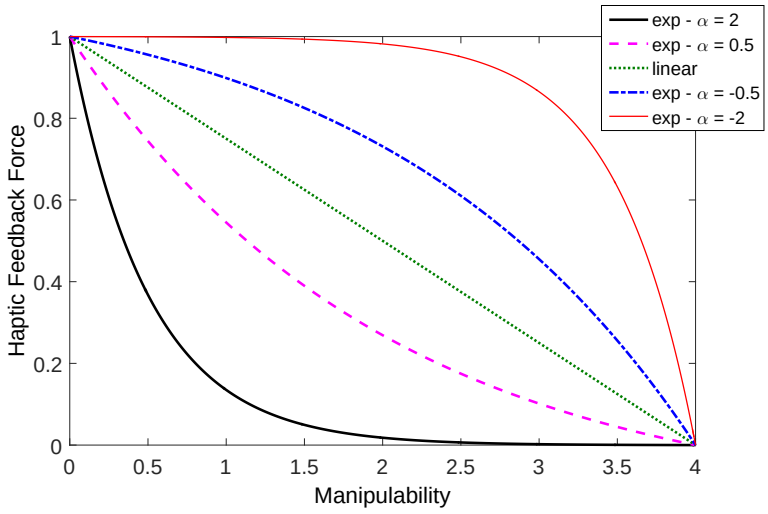


Figure 31. Manipulability vs. haptic force mappings used in user experiments.

that manipulability will have a positive effect on task completion. It will, however, let the operator know how effectively the followers' motions are being controlled by the leader's motion, signifying how easy it is to control the entire team.

Each subject performed five runs, of randomized order, with each run using a unique manipulability-haptic force mapping from the five discussed previously. Each run would start with the robots in their initial configuration, as shown in Figure 28. The subject decided which target location to direct the leader to first and used the haptic device to direct the motion of the leader, and thus the swarm, to this location. When the subject decided that the leader was close enough to the first target location, the subject directed the leader, via the haptic device, to the second target location. When the subject decided that the leader was close enough to the second target location, the run would end. See Figure 32 for of the robot team getting close to one of the target locations. The robots were then reset to their initial configuration before the next run would start. After each consecutive run, data was collected and the subject filled out a NASA Task Load Index (TLX) survey, which measures the difficulty of the task.

Using the position and time data collected during the experiments, several measures were computed for comparison. In order to measure how successful the users were in

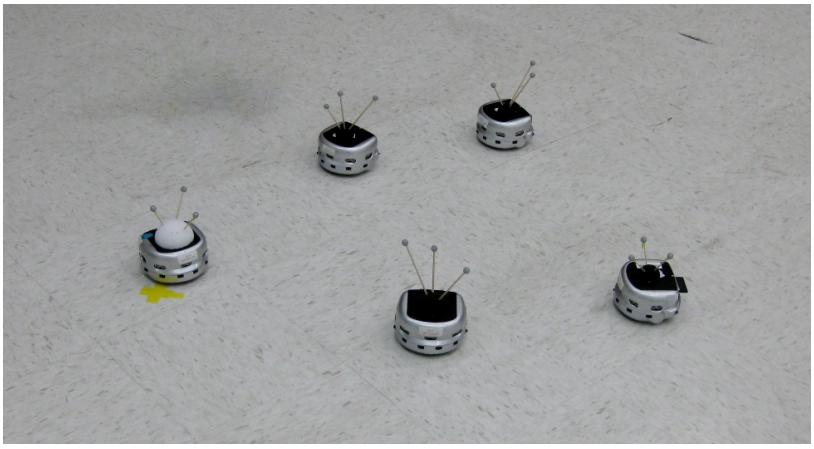


Figure 32. Photo of the robot team approaching one of the target locations during one of the user experiments.

completing the tasks, the shortest distance between the leader and each target location was computed. This distance was computed for each of the two targets using

$$D_k = \min_{t \geq 0}(\|x_l(t) - \tau_k\|), \tag{73}$$

where x_l is the (2-dimensional) position of the leader and τ_k is the (static) position of the k'th target location, $k \in 1, 2$.

A value of zero for D_k means that the leader was precisely on top of the 'X' at target location k sometime during a run, whereas greater values indicate that the subject performing the experiment never reached the target location exactly. Smaller values indicate that the subject was able to drive the leader closer to the target location. In addition to the distance from the target locations, the total time that it took to complete both tasks was computed. For purposes of this measure, task completion is defined as the leader being within 15 cm of the second target location. By averaging these measures across the ten sets of data, it was found that the exponential mapping with $\alpha = 0.5$ led to both the shortest task completion time and the shortest distance to target location, for both of the locations. This can be seen in Table 1.

The last objective measure that was computed and analyzed was the average manipulability of the robot team throughout each of the users' five runs. The ten sets of data were

Table 1. Average task completion time and shortest distance to task location for each manipulability-haptic force mapping (with standard deviations in parentheses).

	Manipulability - Force Mapping				
	α = **2**	α = **0.5**	**linear**	α = **-0.5**	α = **-2**
Time (sec)	85.957 (29.553)	84.391 (23.489)	115.380 (56.554)	87.991 (23.880)	87.160 (17.796)
D_1 **(cm)**	4.300 (2.481)	2.680 (1.371)	3.754 (3.152)	3.133 (1.882)	4.593 (1.833)
D_2 **(cm)**	5.047 (3.735)	3.574 (2.885)	6.745 (6.597)	7.581 (5.523)	5.977 (4.246)

Table 2. Average manipulability over entire run for each mapping between manipulability and haptic force (with standard deviations in parentheses).

	Manipulability - Force Mapping				
	α = **2**	α = **0.5**	**linear**	α = **-0.5**	α = **-2**
$\tilde{M}$	2.6781 (0.3069)	2.5646 (0.1976)	2.5695 (0.1754)	2.5890 (0.2800)	2.5997 (0.2029)

averaged to see how the average manipulability was affected by the haptic mapping and these values can be found in Table 2. It is expected that the mapping with α = -2 would be most likely to force the user to go in directions with higher manipulability since the forces are higher under this mapping. Since the mapping with α = 2 gives off the lowest resistive haptic forces, it is expected that a high manipulability wouldn't be maintained under this mapping. However, the results did not reflect this intuition. The mapping with α = 2 gave the highest average manipulability of all of the mappings. The other four mappings all had very similar manipulability values.

This may be due to the nature of the tasks and the initial configuration of the robots. If users were most inclined to move the leader in a direction that happened to have a high manipulability to begin with, then the addition of stronger forces wouldn't make much of a difference. In addition, some users were fixated on the directions that they wanted to move the leader and were unwilling to let the haptic forces influence their decisions. Perhaps stronger forces would discourage this behavior, but the forces most likely cannot

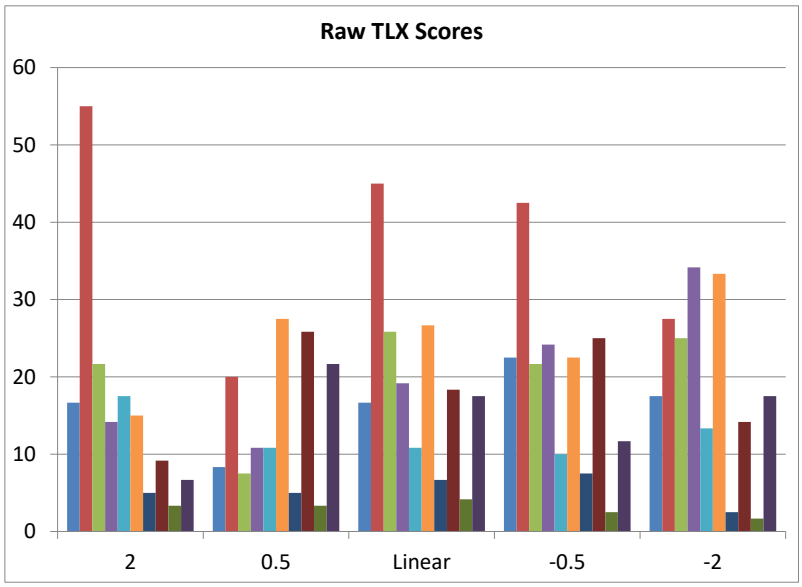


Figure 33. Raw TLX scores given by the subjects for each of the five manipulability-force mappings. For each manipulability-haptic force mapping on the horizontal axis (exponential mapping with α = 2, 0.5, -0.5, -2, and linear mapping), the ten bars represent the TLX scores given by each of the ten subjects during the user studies. The TLX scores range from 0-100 where 0 represents a low workload and 100 represents a high workload.

be high enough to impede motion completely due to the limitations of the PHANTOM Omni device. However, these observations are merely speculative and are not supported by the data collected.

The TLX survey required the human subjects to rate each task from 0-100 on six scales: mental demand, physical demand, temporal demand, performance, effort, and frustration. These six measures were averaged to produce a raw TLX score, where a lower number represents a lower workload required for the task. Each of the ten subjects filled out the survey five times, once for each of the five different manipulability-force mappings. The raw TLX scores can be seen in Figure 33.

The mean and standard variation of the raw TLX scores for each of the five mappings were computed and are given in Table 3 and depicted in Figure 34. From these results, it can be seen that the exponential mapping with $\alpha = 0.5$ produced the best results in terms of workload. It should be noted that the standard deviation values are a bit high and could be

Table 3. Mean NASA TLX scores for each of the five manipulability-force mappings (with standard deviations in parentheses).

	Manipulability - Force Mapping				
	α = 2	α = 0.5	linear	α = -0.5	α = -2
TLX Score	16.4167 (14.8087)	14.0833 (8.8581)	19.0833 (11.6829)	19.0000 (11.4786)	18.6666 (11.4274)

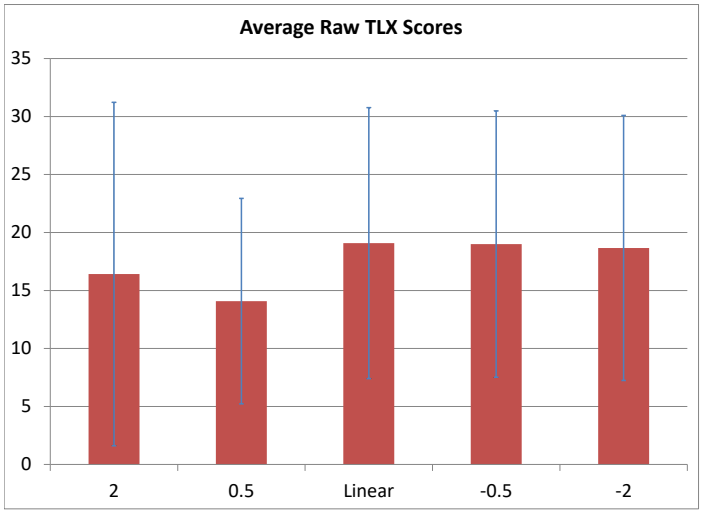


Figure 34. Average of the raw TLX scores given by the subjects for each of the five manipulability-force mappings. The lines above and below the bars show one standard deviation of the data.

due to the fact that the NASA TLX score measures perceived workload, which may vary from person to person.

Based on these results, an exponential mapping with α = 0.5 outperformed the other four mappings in terms of easiness of task and task completion. This particular mapping provides feedback to the user without providing so much force that it makes the task difficult to complete, which may be the case in the α = -0.5 and α = -2 mappings. Since the goal was to move the leader to a specific set of locations, the user must balance his or her desire to move in a certain direction (towards the task location) with the feedback that the haptic controller is giving. Nonetheless, the haptic device provides enough feedback that the user has some intuition about what is going on with the system internally.

6.6 Conclusions

In this chapter, we presented the notion of approximate manipulability in leader-follower networks and relayed this metric via force feedback in order to assist a human operator in controlling a robot swarm with a haptic device. Different mappings between the approximate manipulability of the network and the haptic feedback force were explored. It was found that an exponential mapping with a parameter of $\alpha = 0.5$ is preferable over the other exponential mappings as well as the linear mapping. The experimental results show that the choice of mapping played a large role in both robot team performance and workload.

CHAPTER 7

CONCLUSIONS

In this thesis, we presented models that couple human opinion dynamics and standard multi-agent control laws in order to analyze the performance of human-robot teams. The main contribution of this work was the development of two trust models – self-centered and team-oriented – and the coupling between the trust metric and the evolution of the agents' states. We analyzed the coupled systems to determine how trust affects the performance of human-robot teams and gave conditions under which these systems achieve their desired performance. We validated this work by showing our results are psychologically-consistent in that the system exhibits belief polarization, group polarization, and a positive trust-performance correlation.

We additionally presented two related extensions to this work. One such extension addressed the energy constraints that are often present in multi-robot scenarios. In this work, an algorithm was developed to allow a group of robots to rendezvous in the shortest amount of time while utilizing the least amount of energy and it was shown that the meeting point is affected by the fact that robots initially have different battery levels. The second extension explored more aspects of human-swarm interaction, this time using experimental user studies. Manipulability was presented as a useful property to be relayed to a human operator controlling a group of robots, and it was fed back via a haptic device that the operator used to control the velocity of a leader robot. User studies were done to analyze different mappings between manipulability and the haptic force on the device.

REFERENCES

[1] A. Freedy, E. DeVisser, G. Weltman, and N. Coeyman, "Measurement of trust in human-robot collaboration," in *International Symposium on Collaborative Technologies and Systems*, pp. 106–114, May 2007.

[2] J. Leitner, "Multi-robot cooperation in space: A survey," in *Advanced Technologies for Enhanced Quality of Life*, pp. 144–151, July 2009.

[3] C. J. R. McCook and J. M. Esposito, "Flocking for heterogeneous robot swarms: A military convoy scenario," in *Southeastern Symposium on System Theory*, pp. 26–31, 2007.

[4] J. L. Baxter, E. K. Burke, J. M. Garibaldi, and M. Norman, *Multi-Robot Search and Rescue: A Potential Field Based Approach*, pp. 9–16. Springer Berlin Heidelberg, 2007.

[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[6] J. Lin, A. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem," in *IEEE Conf. Decision and Control*, vol. 2, pp. 1508–1513 Vol.2, Dec 2003.

[7] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, pp. 243–255, April 2004.

[8] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton University Press, 2009.

[9] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 905–908, 2001.

[10] J. A. Fax and R. M. Murray, "Graph Laplacians and stabilization of vehicle formations," *Proc. IFAC World Congress*, pp. 283–288, 2002.

[11] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[12] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa, "Containment control in mobile networks," *IEEE Trans. Autom. Control*, vol. 53, Sept 2008.

[13] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, pp. 9–26, Feb 2016.

[14] J. Nagi, A. Giusti, L. M. Gambardella, and G. A. D. Caro, "Human-swarm interaction using spatial gestures," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3834–3841, Sept 2014.

[15] T. Setter, A. Fouraker, H. Kawashima, and M. Egerstedt, "Haptic interactions with multi-robot swarms using manipulability," *Journal of Human-Robot Interaction*, vol. 4, no. 1, 2015.

[16] A. Kolling, S. Nunnally, and M. Lewis, "Towards human control of robot swarms," in *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 89–96, March 2012.

[17] N. Wang, D. V. Pynadath, and S. G. Hill, "Trust calibration within a human-robot team: Comparing automatically generated explanations," in *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 109–116, March 2016.

[18] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. C. Chen, E. J. de Visser, and R. Parasuraman, "A meta-analysis of factors affecting trust in human-robot interaction," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 53, no. 5, pp. 517–527, 2011.

[19] M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco, "Impact of robot failures and feedback on real-time trust," in *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 251–258, March 2013.

[20] J. Y. C. Chen and M. J. Barnes, "Human-agent teaming for multirobot control: A review of human factors issues," *IEEE Transactions on Human-Machine Systems*, vol. 44, pp. 13–29, Feb 2014.

[21] C. E. Harriott, A. E. Seiffert, S. T. Hayes, and J. A. Adams, "Biologically-inspired human-swarm interaction metrics," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 58, no. 1, pp. 1471–1475, 2014.

[22] X. Wang, Z. Shi, F. Zhang, and Y. Wang, "Mutual trust based scheduling for (semi)autonomous multi-agent systems," in *American Control Conference*, pp. 459–464, July 2015.

[23] H. Saeidi, F. McLane, B. Sadrfaidpour, E. Sand, S. Fu, J. Rodriguez, J. R. Wagner, and Y. Wang, "Trust-based mixed-initiative teleoperation of mobile robots," in *American Control Conference*, pp. 6177–6182, July 2016.

[24] F. Gao, A. S. Clare, J. C. Macbeth, and M. Cummings, "Modeling the impact of operator trust on performance in multiple robot control," in *AAAI Spring Symposium*, AAAI, 2013.

[25] D. Acemoglu and A. Ozdaglar, "Opinion dynamics and learning in social networks," *Dynamic Games and Applications*, vol. 1, no. 1, pp. 3–49, 2010.

[26] H. Xia, H. Wang, and Z. Xuan, "Opinion dynamics: A multidisciplinary review and perspective on future research," *Int. J. Knowl. Syst. Sci.*, vol. 2, pp. 72–91, Oct 2011.

[27] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Reviews of modern physics*, vol. 81, no. 2, p. 591, 2009.

[28] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch, "Mixing beliefs among interacting agents," *Advances in Complex Systems*, vol. 3, pp. 87–98, 2001.

[29] R. Hegselmann and U. Krause, "Opinion dynamics and bounded confidence: models, analysis and simulation," *The Journal of Artificial Societies and Social Simulation*, vol. 5, 2002.

[30] C. Altafini, "Consensus problems on networks with antagonistic interactions," *IEEE Trans. Autom. Control*, vol. 58, pp. 935–946, April 2013.

[31] C. Altafini and G. Lini, "Predictable dynamics of opinion forming for networks with antagonistic interactions," *IEEE Trans. Autom. Control*, vol. 60, pp. 342–357, Feb 2015.

[32] A. Proskurnikov, A. Matveev, and M. Cao, "Consensus and polarization in altafini's model with bidirectional time-varying network topologies," in *IEEE Conf. Decision and Control*, 2014.

[33] A. Proskurnikov, A. Matveev, and M. Cao, "Opinion dynamics in social networks with hostile camps: Consensus vs. polarization," *IEEE Trans. Autom. Control*, vol. PP, no. 99, pp. 1–1, 2015.

[34] W. Xia, M. Cao, and K. H. Johansson, "Structural balance and opinion separation in trust-mistrust social networks," *IEEE Transactions on Control of Network Systems*, vol. 3, pp. 46–56, March 2016.

[35] G. Theodorakopoulos and J. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 318–328, Feb 2006.

[36] Y. Wang and J. Vassileva, "Bayesian network-based trust model," in *IEEE/WIC International Conf. on Web Intelligence*, 2003.

[37] M. Virendra, M. Jadliwala, M. Chandrasekaran, and S. Upadhyaya, "Quantifying trust in mobile ad-hoc networks," in *International Conf. on Integration of Knowledge Intensive Multi-Agent Systems*, 2005.

[38] U. Krause, "A discrete nonlinear and non-autonomous model of consensus formation," in *Proc. Commun. Difference Equations*, pp. 227–236, Gordon and Breach Pub, 2000.

[39] J. Lorenz, "A stabilization theorem for dynamics of continuous opinions," *Physica A: Statistical Mechanics and its Applications*, vol. 355, no. 1, pp. 217 – 223, 2005.

[40] J. Lorenz, "Consensus strikes back in the hegselmann-krause model of continuous opinion dynamics under bounded confidence," *J. Artif. Soc. Soc. Simulat.*, vol. 9, no. 1, p. 8, 2006.

[41] V. Blondel, J. Hendrickx, and J. Tsitsiklis, "On krause's multi-agent consensus model with state-dependent connectivity," *IEEE Trans. Autom. Control*, vol. 54, pp. 2586–2597, 2009.

[42] V. D. Blondel, J. M. Hendrickx, and J. N. Tsitsiklis, "On krause's consensus formation model with state-dependent connectivity," *CoRR*, 2008.

[43] X. Liu and J. Baras, "Using trust in distributed consensus with adversaries in sensor and other networks," in *International Conference on Information Fusion*, pp. 1–7, July 2014.

[44] D. G. Mikulski, F. L. Lewis, E. Y. Gu, and G. R. Hudas, "Trust method for multi-agent consensus," in *Proc. SPIE*, vol. 8387, 2012.

[45] A. Pierson and M. Schwager, "Adaptive inter-robot trust for robust multi-robot sensor coverage," in *International Symposium of Robotics Research (ISRR)*, Dec 2013.

[46] C. Pippin and H. Christensen, "Trust modeling in multi-robot patrolling," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 59–66, May 2014.

[47] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[48] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, pp. 1520–1533, Sept 2004.

[49] T. Setter, A. Gasparri, and M. Egerstedt, "Trust-based interactions in teams of mobile agents," in *American Control Conference*, 2016.

[50] T. Setter, A. Gasparri, and M. Egerstedt, "Trust in multi-agent networks: From self-centered to team-oriented," in *American Control Conference*, May 2017. To appear.

[51] T. Setter, A. Gasparri, and M. Egerstedt, "A psychologically consistent trust notion for multi-agent coordination." Submitted to IEEE Transactions on Control of Network Systems.

[52] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Trans. Robot. Autom.*, vol. 15, pp. 818–828, Oct 1999.

[53] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

[54] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[55] P. Ogren, M. Egerstedt, and X. Hu, "A control lyapunov function approach to multi-agent coordination," *IEEE Trans. Robot. Autom.*, vol. 18, pp. 847–851, Oct 2002.

[56] R. Olfati-Saber and R. Murray, "Distributed structural stabilization and tracking for formations of dynamic multi-agents," in *IEEE Conf. Decision and Control*, 2002.

[57] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, p. 118121, 1974.

[58] N. E. Friedkin, "The problem of social control and coordination of complex systems in sociology: A look at the community cleavage problem," *IEEE Control Systems*, vol. 35, pp. 40–51, June 2015.

[59] C. Lord, L. Ross, and M. Lepper, "Biased assimilation and attitude polarization: The effects of prior theories on subsequently considered evidence," *J. Pers. Soc. Psychol.*, vol. 37, pp. 2098–2109, 1979.

[60] P. Dandekar, A. Goel, and D. Lee, "Biased assimilation, homophily and the dynamics of polarization," *CoRR*, vol. abs/1209.5998, 2012.

[61] A. Flache and M. W. Macy, "Small worlds and cultural polarization," *The Journal of Mathematical Sociology*, vol. 35, no. 1-3, pp. 146–176, 2011.

[62] F. Erdem and J. Ozen, "Cognitive and affective dimensions of trust in developing team performance," *Team Performance Management: An International Journal*, vol. 9, pp. 131–135, 2003.

[63] E. Aronson, T. Wilson, and R. Akert, *Social Psychology*. Prentice Hall, 2010.

[64] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Trans. Autom. Control*, vol. 48, pp. 692–697, April 2003.

[65] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectedness preserving distributed swarm aggregation for multiple kinematic robots," *IEEE Transactions on Robotics*, vol. 24, pp. 1213–1223, Oct 2008.

[66] A. Gasparri, G. Oriolo, A. Priolo, and G. Ulivi, "A swarm aggregation algorithm based on local interaction for multi-robot systems with actuator saturations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 539–544, Oct 2012.

[67] W. Li and M. W. Spong, "Unified cooperative control of multiple agents on a sphere for different spherical patterns," *IEEE Trans. Autom. Control*, vol. 59, pp. 1283–1289, May 2014.

[68] V. Gazi and K. M. Passino, *Swarm Stability and Optimization*. Springer Publishing Company, Incorporated, 1st ed., 2011.

[69] V. Srivastava and N. E. Leonard, "Collective decision-making in ideal networks: The speed-accuracy trade-off," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 121–132, 2014.

[70] R. Ratcliff and G. McKoon, "The diffusion decision model: Theory and data for two-choice decision tasks," *Neural Computation*, vol. 20, no. 4, pp. 873–922, 2008.

[71] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, and J. D. Cohen, "The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks," *Psychological Review*, vol. 113 (4), pp. 700 – 765, 2006.

[72] I. Newton, "Chapter 4 - weather effects and other aspects," in *The Migration Ecology of Birds* (I. Newton, ed.), pp. 67 – 94, Oxford: Academic Press, 2007.

[73] F. McKinney, *Ecology and Management of Breeding Waterfowl*, ch. Courtship, Pair Formation, and Signal Systems, pp. 214–250. University of Minnesota Press, 1992.

[74] W. Cresswell, "Flocking is an effective anti-predation strategy in redshanks, tringa totanus," *Animal Behaviour*, vol. 47, no. 2, pp. 433 – 442, 1994.

[75] J. L. Mishra, "Factors affecting group decision making: an insight on information practices by investigating decision making process among tactical commanders," in *Proceedings of ISIC: the information behaviour conference, Leeds*, 2014.

[76] S. Keegan, *The Psychology of Fear in Organizations: How to Transform Anxiety into Well-being, Productivity and Innovation*. Kogan Page, 2015.

[77] T. Setter and M. Egerstedt, "Minimum time power-aware rendezvous for multi-agent networks," in *IEEE Conference on Control Applications (CCA)*, pp. 2159–2164, Oct 2014.

[78] T. Setter and M. Egerstedt, "Energy-constrained coordination of multi-robot teams," *IEEE Trans. Control Syst. Technol.*, vol. PP, no. 99, pp. 1–7, 2016.

[79] A. Shia, F. Bastani, and I.-L. Yen, "A highly resilient framework for autonomous robotic swarm systems operating in unknown, hostile environments," in *International Symposium on Autonomous Decentralized Systems (ISADS)*, pp. 147–153, March 2011.

[80] A. Alfieri, A. Bianco, P. Brandimarte, and C. Chiasserini, "Maximizing system lifetime in wireless sensor networks," *European Journal of Operational Research*, vol. 181, no. 1, pp. 390 – 402, 2007.

[81] M. Sarkar and R. Cruz, "An adaptive "sleep" algorithm for efficient power management in wlans," in *IEEE 61st Vehicular Technology Conference*, vol. 3, May 2005.

[82] R. Subramanian and F. Fekri, "Sleep scheduling and lifetime maximization in sensor networks: fundamental limits and optimal solutions," in *International Conference on Information Processing in Sensor Networks*, pp. 218–225, 2006.

[83] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Peas: a robust energy conserving protocol for long-lived sensor networks," in *IEEE International Conference on Network Protocols*, 2002.

[84] P. Martin, R. Galvan-Guerra, M. Egerstedt, and V. Azhmyakov, "Power-aware sensor coverage: An optimal control approach," in *International Symposium on Mathematical Theory of Networks and Systems*, 2010.

[85] H. Jaleel, A. Rahmani, and M. Egerstedt, "Probabilistic lifetime maximization of sensor networks," *IEEE Trans. Autom. Control*, vol. 58, pp. 534–539, 2013.

[86] H. Jaleel, S. Bopardikar, and M. Egerstedt, "Towards power-aware rendezvous," in *IEEE Conf. Decision and Control*, 2011.

[87] H. Jaleel, Y. Wardi, and M. Egerstedt, "Minimizing mobility and communication energy in robotic networks: An optimal control approach," in *American Control Conference*, 2014.

[88] Y. Yan and Y. Mostofi, "Efficient communication-aware dynamic coverage using space-filling curves," in *American Control Conference*, June 2014.

[89] H. Jaleel and M. Egerstedt, "Power-aware rendezvous with shrinking footprints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2772–2777, Sept 2011.

[90] Z. Meng, W. Ren, Y. Cao, and Z. You, "Leaderless and leader-following consensus with communication and input delays under a directed network topology," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 41, pp. 75–88, Feb 2011.

[91] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Trans. Autom. Control*, vol. 53, pp. 1503–1509, July 2008.

[92] W. Yu, G. Chen, M. Cao, and J. Kurths, "Second-order consensus for multiagent systems with directed topologies and nonlinear dynamics," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 40, pp. 881–891, June 2010.

[93] J. Qin, W.-X. Zheng, and H. Gao, "Coordination of multiple agents with double-integrator dynamics under generalized interaction topologies," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 42, pp. 44–57, Feb 2012.

[94] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal velocity profiles for car-like robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1457–1462, May 2011.

[95] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2011.

[96] D. Feijer and F. Paganini, "Stability of primal-dual gradient dynamics and applications to network optimization," *Automatica*, vol. 46, pp. 1974–1981, Dec. 2010.

[97] T. Setter, H. Kawashima, and M. Egerstedt, "Team-level properties for haptic human-swarm interactions," in *American Control Conference*, pp. 453–458, July 2015.

[98] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, and B. Schmidt, "Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots," in *AAAI Spring Symposium*, March 2006.

[99] M. Cummings, "Human supervisory control of swarming networks," in *Autonomous Intelligent Networked Systems Conference*, June 2004.

[100] R. Arkin and K. Ali, "Integration of reactive and telerobotic control in multi- agent robotic systems," in *Third International Conference on Simulation of Adaptive Behavior*, pp. 473–478, 1994.

[101] A. Atherton and M. Goodrich, "Supporting remote and mobile manipulation with an ecological augmented virtuality interface," in *Proceedings of AISB-HRI Symposium – New Frontiers in Human-Robot Interaction*, 2009.

[102] Z. Kira and M. A. Potter, "Exerting human control over decentralized robot swarms," in *Proceedings of International Conference on Autonomous Robots and Agents*, 2009.

[103] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Transactions on Robotics*, vol. 31, pp. 489–493, April 2015.

[104] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt., "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.

[105] J. de la Croix and M. Egerstedt, "Controllability characterizations of leader-based swarm interactions," in *AAAI Symposium on Human Control of Bio-Inspired Swarms*, 2012.

[106] T. Yoshikawa, "Manipulability of robotic mechanisms," in *The International Journal of Robotics Research*, vol. 4, pp. 3–9, 1985.

[107] A. Bicchi, C. Melchiorri, and D. Balluchi, "On the mobility and manipulability of general multiple limb robots," in *IEEE Trans. Robot. Autom.*, vol. 11, pp. 215–228, 1995.

[108] A. Bicchi and D. Prattichizzo, "Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms," in *IEEE Trans. Robot. Autom.*, vol. 16, pp. 336–345, 2000.

[109] H. Kawashima and M. Egerstedt, "Approximate manipulability of leader-follower networks," in *IEEE Conf. Decision and Control*, pp. 6618–6623, 2011.

[110] H. Kawashima and M. Egerstedt, "Manipulability of leader-follower networks with the rigid-link approximation," *Automatica*, 2014.

[111] M. Riedel, A. Franchi, P. R. Giordano, H. H. Bülthoff, and H. I. Son, "Experiments on intercontinental haptic control of multiple uavs," in *Intelligent Autonomous Systems 12*, pp. 227–238, Springer, 2013.

[112] D. Lee, O. Martinez-Palafox, and M. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Application," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 366–371, April 2005.

[113] D. Lee and M. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Theory," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 360–365, April 2005.

[114] E. Rodriguez-Seda, J. Troy, C. Erignac, P. Murray, D. Stipanovic, and M. Spong, "Bilateral teleoperation of multiple mobile agents: Coordinated motion and collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 18, pp. 984–992, July 2010.

[115] C. Secchi, A. Franchi, H. Bulthoff, and P. Robuffo Giordano, "Bilateral control of the degree of connectivity in multiple mobile-robot teleoperation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3645–3652, May 2013.

[116] S. Nunnally, P. Walker, M. Lewis, N. Chakraborty, and K. Sycara, "Using haptic feedback in human robotic swarms interaction.," in *Proceedings of the Human Factors and Ergonomics Society*, pp. 1047–1051, 2013.

[117] M. Ishihara, S. Suzuki, K. Ohshima, and J. Shirataki, "Evaluation of force delays on the operation of haptic sense," in *International Conference on Control, Automation and Systems*, pp. 2049–2053, Oct 2008.

[118] B. Knorlein, M. Di Luca, and M. Harders, "Influence of visual and haptic delays on stiffness perception in augmented reality," in *IEEE International Symposium on Mixed and Augmented Reality*, pp. 49–52, Oct 2009.